

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**  
**Навчально-науковий інститут економіки, управління, права та**  
**інформаційних технологій**  
**Кафедра інформаційних систем та технологій**

# **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення інтерфейсу користувача інтернет-магазину  
спортивних товарів»**

Виконав: здобувач вищої освіти  
за освітньою програмою  
Інформаційні управляючі системи  
спеціальності 126 Інформаційні  
системи та технології  
ступеня вищої освіти бакалавр  
групи 126ІСТ\_бд\_2022  
Мироненко Михайло Іванович  
Керівник: Копішинська Олена  
Петрівна  
Рецензент: Муравльов Володимир  
В'ячеславович

**Полтава – 2026 року**

**ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**  
**Навчально-науковий інститут економіки, управління, права та**  
**інформаційних технологій**  
**Кафедра інформаційних систем та технологій**

Освітня програма Інформаційні управляючі системи  
Спеціальність 126 Інформаційні системи та технології  
Рівень вищої освіти перший (бакалаврський)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

\_\_\_\_\_ Юрій УТКІН

«10» червня 2025 року

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

**Мироненка Михайла Івановича**

1. Тема кваліфікаційної роботи:  
«Розроблення інтерфейсу користувача інтернет-магазину спортивних товарів»,  
Керівник роботи: к. ф.-м. н., доцент, професор кафедри інформаційних систем та технологій Копішинська Олена Петрівна.  
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: інформаційні ресурси бібліотек JavaScript React <https://react.dev/>, Redux <https://redux.js.org/>, редактор вихідного коду <https://code.visualstudio.com/>, платформа зберігання даних Firebase <https://firebase.google.com/>, платформа обробки образень Cloudinary <https://cloudinary.com/>, <https://www.figma.com/design/>, статистичні дані <https://mate.academy/>
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):  
Розділ 1. Огляд загальних функцій інтернет-магазинів  
Розділ 2. Формування концепції та обґрунтування методів розробки інтернет-магазину спортивного одягу  
Розділ 3. Практичні аспекти програмної реалізації інтернет-магазину спортивного одягу
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження

## 6. Консультанти розділів кваліфікаційної роботи

| Розділ                                                      | Прізвище, ініціали та посада консультанта                                            | Підпис, дата   |                  |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------|----------------|------------------|
|                                                             |                                                                                      | завдання видав | завдання отримав |
| Оцінювання економічної ефективності результатів дослідження | Дивнич О. Д., к. е. н.,<br>доцент, доцент кафедри економіки та публічного управління | 01.04.2026 р.  | 29.05.2026 р.    |

## 7. Дата видачі завдання «10» червня 2025 року

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи                                                              | Строк виконання етапів кваліфікаційної роботи | Примітка |
|-------|----------------------------------------------------------------------------------|-----------------------------------------------|----------|
| 1     | Вибір і затвердження теми роботи                                                 | 02.06.2025 р. – 04.06.2025                    |          |
| 2     | Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу | 05.06.2025 р. – 10.06.2025 р.                 |          |
| 3     | Опрацювання джерел інформації                                                    | 11.06.2025 р. – 15.06.2025 р.                 |          |
| 4     | Збір, вивчення і обробка інформації, необхідної для виконання роботи             | 16.06.2025 р. – 20.06.2025 р.                 |          |
| 5     | Виконання теоретико-методологічного розділу роботи                               | 08.09.2025 р. – 01.11.2025 р.                 |          |
| 6     | Виконання дослідницько-аналітичного розділу роботи                               | 19.01.2026 р. – 14.02.2026 р.                 |          |
| 7     | Виконання проєктно-рекомендаційного розділу роботи                               | 16.02.2026 р. – 31.03.2026 р.                 |          |
| 8     | Оцінювання економічної ефективності результатів дослідження                      | 01.04.2026 р. – 29.05.2026 р.                 |          |
| 9     | Оформлення тексту роботи                                                         | 01.06.2026 р. – 06.06.2026р.                  |          |
| 10    | Попередній захист роботи на кафедрі                                              | 08.06.2026 р.                                 |          |
| 11    | Доопрацювання роботи з урахуванням зауважень і пропозицій                        | 09.06.2026 р. – 10.06.2026 р.                 |          |
| 12    | Нормоконтроль                                                                    | 11.06.2026 р. – 15.06.2026 р.                 |          |
| 13    | Захист кваліфікаційної роботи                                                    | 16.06.2026 р. - 19.06.2026 р.                 |          |

**Здобувач вищої освіти**

**Михайло МИРОНЕНКО**

**Керівник роботи**

**Олена КОПШИНСЬКА**

## ЗМІСТ

|                                                                                                                                     |    |
|-------------------------------------------------------------------------------------------------------------------------------------|----|
| РОЗДІЛ 1. ОГЛЯД ЗАГАЛЬНИХ ФУНКЦІЙ ІНТЕРНЕТ-МАГАЗИНІВ .....                                                                          | 8  |
| 1.1 Передумови виникнення та розвиток інтернет-магазинів .....                                                                      | 8  |
| 1.2 Опис існуючих методів розробки інтернет-магазинів .....                                                                         | 11 |
| 1.3 Аналіз конкурентів на ринку інтернет-магазинів спортивного одягу .....                                                          | 18 |
| РОЗДІЛ 2. ФОРМУВАННЯ КОНЦЕПЦІЇ ТА ОБГРУНТУВАННЯ МЕТОДІВ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ СПОРТИВНОГО ОДЯГУ .....                          | 23 |
| 2.1 Складання технічних характеристик функціональних вимог інтернет-магазину спортивного одягу. Визначення цільової аудиторії ..... | 23 |
| 2.2 Підбір та характеристика технологій для розробки інтернет-магазину ....                                                         | 27 |
| 2.3 Розробка макетів окремих сторінок інтернет-магазину .....                                                                       | 39 |
| РОЗДІЛ 3. ПРАКТИЧНІ АСПЕКТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНТЕРНЕТ-МАГАЗИНУ СПОРТИВНОГО ОДЯГУ .....                                         | 42 |
| 3.1 Технічні прийоми створення головної сторінки .....                                                                              | 42 |
| 3.2 Обробка даних на сайті .....                                                                                                    | 47 |
| 3.3 Перевірка та опис економічної ефективності розробки .....                                                                       | 51 |
| ВИСНОВКИ .....                                                                                                                      | 56 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                                                                                                    | 58 |
| ДОДАТКИ .....                                                                                                                       | 62 |

## ВСТУП

Сучасні інформаційні технології істотно змінюють підходи до організації торгівлі, надаючи можливість здійснювати покупки дистанційно, швидко та зручно. Онлайн-магазини дозволяють компаніям розширювати ринки збуту, оптимізувати витрати на утримання фізичних торгових точок і забезпечувати клієнтам доступ до широкого асортименту товарів незалежно від місця перебування.

*Актуальність* теми кваліфікаційної роботи з розроблення оригінального інтерфейсу користувача інтернет-магазину спортивного одягу зумовлена активним розвитком електронної комерції та зростанням популярності здорового способу життя у світі: зростає кількість людей, які займаються спортом, фітнесом або ведуть активний спосіб життя. Інтернет-магазини спортивного одягу стали дуже актуальним варіантом завдяки своїй зручності, широкому асортименту, конкурентоспроможним цінам. Стрімкий розвиток технологій і можливостей доступу до інтернету зробили інтернет-магазини спортивного одягу кращим вибором для спортсменів і любителів спорту. Для таких споживачів важливими є швидкий доступ до інформації про товари, можливість порівняння моделей, ознайомлення з характеристиками та відгуками інших користувачів. Інтернет-магазини створюють сприятливі умови для реалізації цих потреб, поєднуючи зручний інтерфейс, системи пошуку та фільтрації товарів, а також інтеграцію із сучасними платіжними та логістичними сервісами.

*Мета* кваліфікаційної роботи – розробка актуального інтерфейсу інтернет-магазину спортивного одягу, що забезпечує зручне та ефективне використання платформи електронної торгівлі спортивним одягом для зацікавлених користувачів.

*Завдання* кваліфікаційної роботи:

- аналіз ринку інтернет-торгівлі спортивними товарами для визначення основних потреб та очікувань покупців;
- розроблення інтерфейсу користувача інтернет-магазину;

- розроблення панелі управління інтернет-магазину;
- створення системи обробки замовлень;
- проведення тестування, оптимізації та оцінка ефективності розробленого інтернет-магазину.

*Об'єкт дослідження* – розроблення інтернет-магазину спортивного одягу.

*Предмет дослідження* – програмно-технічні засоби для аналізу, розробки та впровадження значущих елементів інтернет-магазину спортивного одягу.

*Методи досліджень*: аналітичний, інформаційно-пошуковий, емпіричний, дедуктивний, проєктний, графічний, програмування та розробка вебдодатків, методи внутрішньої SEO-оптимізації програмного продукту, економічне оцінювання ефективності розробки програмного додатку.

*Апробація* результатів дослідження здійснена у вигляді публікації тез доповіді на студентській науковій конференції та науково-практичній конференції за результатами виробничої практики.

*Практична значущість* дослідження полягає у створенні функціонального та зручного інтернет-магазину спортивного одягу, що сприяє розвитку електронної комерції в сфері товарів для спорту, забезпеченні зручності та доступності для клієнтів, розширенні аудиторії та збільшенні продажів, аналізі ринку та конкурентоспроможності, а також покращенні клієнтського досвіду через зручну навігацію, швидку обробку замовлень, безпечні платежі та швидку доставку.

В якості інформаційної бази було використано: інтернет-ресурси, офіційні сайти розробників інформаційних систем, офіційні сайти інтернет-магазинів, наукові публікації по темі дослідження, дані аналітичних компаній.

Структура кваліфікаційної роботи: пояснювальна записка містить вступ, три розділи основної частини, висновки, список використаних джерел та додатки. Загальний обсяг роботи складає 57 сторінок, 7 таблиць, 32 рисунки та список із 36 використаних джерел.

## **РОЗДІЛ 1**

### **ОГЛЯД ЗАГАЛЬНИХ ФУНКЦІЙ ІНТЕРНЕТ-МАГАЗИНІВ**

#### **1. 1 Передумови виникнення та розвиток інтернет-магазинів**

Поява інтернет-магазинів безпосередньо пов'язана з розвитком електронної комерції та поширенням мережі інтернет наприкінці ХХ століття. Перші ідеї дистанційної електронної торгівлі з'явилися ще у 1979 р., коли британський винахідник Michael Aldrich створив систему електронних покупок, що поєднувала телевізор із комп'ютером через телефонну лінію. Така технологія дозволяла здійснювати транзакції між користувачами та компаніями і вважається одним із перших прикладів електронної комерції [1].

У 1990-х роках із появою Всесвітньої павутини (World Wide Web) почали формуватися перші інтернет-магазини у сучасному розумінні. У 1994 році відбулася одна з перших зафіксованих онлайн-покупок: на сайті NetMarket було продано компакт-диск музичного альбому Ten Summoner's Tales виконавця Sting із використанням зашифрованої передачі даних. Ця подія вважається важливим етапом у розвитку електронної комерції [2]. Подальший розвиток інтернет-магазинів відбувався дуже швидко. У 1994–1995 рр. з'явилися перші комерційні онлайн-платформи та спеціалізовані вебсайти для продажу товарів, а також програмне забезпечення (ПЗ) для створення електронних магазинів, зокрема платформа Intershop, розроблена підприємцем Stephan Schambach. У цей період почали формуватися основні принципи електронної торгівлі: використання вебкаталогів товарів, онлайн-оплати та систем безпечних транзакцій.

Таким чином, інтернет-магазини виникли як результат розвитку інформаційних технологій і мережевих сервісів [3]. Концепція інтернет-магазину обертається навколо ідеї здійснення комерційних операцій через інтернет. Це дозволяє компаніям продавати товари або послуги клієнтам без необхідності мати фізичний магазин. Інтернет-магазини суттєво змінили спосіб ведення бізнесу і стали невід'ємною частиною сучасної цифрової економіки.

Свою популярність інтернет-магазини завдячують кільком факторам. По-перше, вони пропонують зручність і доступність. Клієнти можуть здійснювати покупки в будь-який час і в будь-якому місці, без необхідності їхати до фізичного магазину. Така гнучкість особливо приваблива для зайнятих людей, які не мають часу або бажання відвідувати традиційні магазини. Крім того, інтернет-магазини пропонують широкий вибір товарів, що дозволяє клієнтам порівнювати ціни і знаходити унікальні товари, які можуть бути недоступні в місцевих магазинах.

Крім того, інтернет-магазини значно знизили бар'єри для входу на ринок. Створення інтернет-магазину часто є більш економічно вигідним, ніж відкриття фізичного торгового простору. Малі та середні підприємства, фізичні особи-підприємці можуть легко розпочати онлайн-бізнес, охопивши глобальну клієнтську базу без необхідності значних початкових інвестицій.

Інтернет-магазини виконують кілька важливих завдань. Вони дозволяють компаніям демонструвати та просувати свої товари чи послуги за допомогою детальних описів товарів, зображень та відгуків клієнтів. Вони сприяють безпечним онлайн-транзакціям, забезпечуючи безпеку платіжної інформації клієнтів. Інтернет-магазини також надають канали підтримки клієнтів, дозволяючи покупцям отримати допомогу або вирішити проблеми за допомогою різних засобів, таких як живий чат або електронна пошта [3].

Інтернет-магазини мають такі особливості:

1. Інтеграція з цифровою екосистемою. Вона включає інтеграцію з платіжними шлюзами, системами управління запасами, програмним забезпеченням для управління взаємовідносинами з клієнтами (CRM), постачальниками логістичних послуг, а також маркетинговими платформами. Ці інтеграції дозволяють ефективно обробляти замовлення, відстежувати запаси, залучати клієнтів і проводити цільові маркетингові кампанії.

2. Персоналізація на основі даних. Інтернет-магазини мають перевагу в зборі та аналізі даних про клієнтів, що дозволяє компаніям персоналізувати досвід покупок. Завдяки таким методам, як аналіз даних і машинне навчання,



компанії можуть збирати інформацію про вподобання клієнтів, їхню поведінку та історію покупок. Ця інформація може бути використана для рекомендації відповідних продуктів, пропонування персоналізованих знижок і створення цільових маркетингових кампаній, що підвищує задоволеність клієнтів і стимулює продажі [4].

3. Розширення охоплення ринку. Однією з важливих переваг інтернет-магазинів є можливість охопити глобальну клієнтську базу. Підприємства більше не обмежені певною географічною локацією, а можуть продавати товари клієнтам по всьому світу. Таке розширення охоплення ринку відкриває нові можливості для зростання і дозволяє компаніям освоювати нішеві ринки або обслуговувати певні сегменти клієнтів.

4. Багатоканальна присутність. Багато інтернет-магазинів зараз застосовують багатоканальний підхід, інтегруючи свою присутність в інтернеті з фізичними роздрібними магазинами. Це дозволяє клієнтам безперешкодно переходити між онлайн і офлайн-каналами, наприклад, здійснювати покупки онлайн і забирати їх у магазині, або навпаки. Забезпечуючи цілісний досвід покупок різними способами, бізнес може задовольнити різні вподобання клієнтів і підвищити їхню лояльність [5].

5. Конкурентне середовище та очікування клієнтів. Популярність інтернет-магазинів призвела до посилення конкуренції у сфері електронної комерції. Щоб виділитися на переповненому ринку, компаніям необхідно постійно впроваджувати інновації та диференціювати себе. Крім того, змінилися очікування клієнтів, і тепер покупки вимагають швидкої доставки, безпроблемного повернення, безпечних транзакцій та персоналізованого обслуговування. Інтернет-магазини повинні адаптуватися до цих очікувань, щоб залишатися конкурентоспроможними та утримувати клієнтів.

Важливо зазначити, що специфічні особливості та функціональні можливості інтернет-магазинів можуть відрізнятися залежно від використовуваної платформи або програмного забезпечення, галузі та бізнес-моделі. Однак основна суть залишається незмінною: надання цифрової

платформи для бізнесу для продажу товарів або послуг і надання клієнтам можливості зручно здійснювати покупки через інтернет.

Загалом, інтернет-магазини зробили революцію в тому, як працює бізнес, і як споживачі роблять покупки. Вони розширили охоплення ринку, підвищили зручність, сприяли конкуренції та інноваціям. Оскільки інтернет продовжує розвиватися, інтернет-магазини, ймовірно, відіграватимуть дедалі важливішу роль у світовій економіці.

## **1.2 Опис існуючих методів розробки інтернет-магазинів**

Існують різні методи розробки інтернет-магазинів, кожен з яких має свої особливості та критерії вибору. Розглянемо далі кілька найпоширеніших підходів.

1. Платформи електронної комерції. Платформи електронної комерції надають комплексні рішення для створення інтернет-магазинів. Ці платформи, такі як Shopify, BigCommerce і Magento, пропонують низку функцій, включаючи управління каталогом товарів, функціональність кошика для покупок, платіжні шлюзи і системи управління замовленнями. Вони часто постачаються шаблонами та темами, які налаштовуються, що полегшує бізнесу створення та управління інтернет-магазинами без глибоких технічних знань.

2. Системи управління контентом (CMS). для створення інтернет-магазинів можна використовувати такі платформи CMS, як WordPress, Joomla і Drupal. Вони пропонують плагіни або розширення для електронної комерції, такі як WooCommerce для WordPress, які додають функціональність електронної комерції до CMS [6].

Платформи CMS забезпечують гнучкість у дизайні, управлінні контентом та інтеграції зі сторонніми сервісами. Вони підходять для компаній, які вже мають вебсайт, створений на CMS, і хочуть розширити його в напрямку електронної комерції.

3. Індивідуальна розробка. Вона передбачає створення індивідуального вебсайту з нуля відповідно до конкретних бізнес-вимог. Цей метод, зазвичай, передбачає наймання професійного веброзробника або команди розробників, які володіють знаннями у сфері вебдизайну, мов програмування та рішень для електронної комерції. Процес починається з планування та розробки структури, макета та інтерфейсу вебсайту. Потім розробники впроваджують необхідні функції, такі як управління каталогом товарів, кошиком для покупок, платіжними шлюзами та системами обробки замовлень. Індивідуальна розробка пропонує максимальну гнучкість і можливості кастомізації, але вимагає більше часу, ресурсів і технічної експертизи порівняно з іншими методами.

4. Headless Commerce. «Headless Commerce означає відокремлення frontend-додатків електронної комерції від backend [7]. Архітектура, яка відокремлює презентаційний рівень інтерфейсу від внутрішньої функціональності електронної комерції дозволяє компаніям створювати високастомізовані інтерфейси з використанням сучасних технологій, таких як React або Vue.js, одночасно використовуючи спеціалізовані API-інтерфейси електронної комерції або мікросервіси для бекенд-частини.

API (від англ. Application programming interface – інтерфейс програмування додатків) – набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах [8]. «Безголова» комерція пропонує гнучкість, масштабованість і можливість надавати послідовний досвід на різних каналах, таких як веб, мобільні пристрої та пристрої інтернету речей.

5. Аутсорсинг. Компанії можуть віддати розробку своїх інтернет-магазинів на аутсорсинг спеціалізованим агентствам з веброзробки або фрілансерам. Цей варіант дозволяє бізнесу використовувати досвід професіоналів, які спеціалізуються на розробці електронної комерції. Аутсорсинг може бути економічно вигідним рішенням для бізнесу з обмеженими технічними ресурсами або специфічними вимогами до проекту.

Для прикладу розглянемо вебплатформу електронної комерції Shopify.

Shopify – це повноцінна комерційна платформа, яка дозволяє створювати, розвивати та керувати бізнесом [9]. Головні переваги Shopify:

1. Зручний інтерфейс, який дозволяє компаніям легко створювати свої інтернет-магазини. Він надає широкий вибір професійно розроблених тем, які можна налаштувати відповідно до ідентичності бренду. Темі адаптовані для мобільних пристроїв, що забезпечує безперебійний досвід покупок на всіх пристроях. Компанії також можуть налаштовувати дизайн, змінюючи кольори, шрифти і макети або створюючи власні теми за допомогою інструментів розробки тем Shopify.

2. Надійна система управління продуктами, яка дозволяє компаніям створювати та керувати своїм каталогом продуктів. Підтримує різні типи продуктів, включаючи фізичні товари, цифрові завантаження та послуги. Компанії можуть додавати описи продуктів, зображення, ціни та інформацію про відстеження запасів. Shopify також пропонує такі функції, як варіації продуктів, організація продуктів у колекції та інструменти пошукової оптимізації (SEO) для оптимізації видимості продуктів.

3. Безпечна система кошика для покупок, яка дозволяє клієнтам додавати товари до свого кошика, розраховувати вартість доставки та переходити до оформлення замовлення. Він підтримує кілька платіжних шлюзів, включаючи Shopify Payments, що дозволяє компаніям приймати платежі кредитними картками безпосередньо через платформу. Shopify також інтегрується зі сторонніми платіжними шлюзами, надаючи бізнесу гнучкість у виборі найбільш підходящого варіанту для своїх клієнтів.

4. Інтуїтивно зрозуміла система управління замовленнями, яка дозволяє компаніям обробляти замовлення, виставляти рахунки, відстежувати статус замовлення та надсилати клієнтам автоматичні сповіщення про замовлення. Вона надає функції управління запасами, які допомагають відстежувати рівень запасів, налаштовувати автоматичне оновлення та отримувати сповіщення про низький рівень запасів. Shopify інтегрується з транспортними перевізниками для оптимізації виконання замовлень і генерувати транспортні етикетки.

5. Платформа має великий магазин додатків, який пропонує широкий спектр розширень для покращення функціональності інтернет-магазинів.

Компанії можуть вибирати з різних додатків, які надають додаткові функції, такі як маркетинг електронною поштою, інтеграція з соціальними мережами, відгуки клієнтів, відновлення покинутих кошиків і розширена аналітика. Магазин додатків дозволяє компаніям налаштовувати свої інтернет-магазини відповідно до конкретних потреб та галузевих вимог.

6. Shopify – це хмарна платформа, що означає, що компаніям не потрібно турбуватися про хостинг або управління серверами. Платформа забезпечує надійну продуктивність, швидке завантаження сторінок і легко справляється зі стрибками трафіку. Shopify створений для роботи з компаніями будь-якого розміру, від невеликих стартапів до підприємств, забезпечуючи масштабованість і здатність рости разом з бізнесом.

7. Цілодобова підтримка клієнтів через різні канали, включаючи чат, електронну пошту та телефон. Він надає вичерпну документацію, навчальні посібники та інструкції, які допомагають бізнесу орієнтуватися та максимально використовувати функції платформи. Shopify також має активний форум спільноти, де користувачі можуть спілкуватися, ділитися досвідом та шукати порад від колег-підприємців.

Серед головних недоліків Shopify можна назвати кілька пунктів.

1. Структура витрат працює за моделлю, що базується на підписці. Це означає, що підприємства повинні сплачувати щомісячну плату за використання платформи. Існують різні тарифні плани, але витрати можуть суттєво впливати на бюджет, особливо для малого бізнесу або стартапів. При використанні сторонніх платіжних шлюзів замість Shopify Payments може стягуватися комісія за транзакції, що може додатково вплинути на загальні витрати.

2. Обмеження кастомізації. Хоча Shopify пропонує широкий вибір тем і варіантів дизайну, але можуть існувати обмеження щодо повної свободи дизайну. Для налаштування зовнішнього вигляду та функціональності певних елементів можуть знадобитися знання кодування або допомога розробника.

Компанії з унікальними вимогами до дизайну або складними потребами в кастомізації можуть виявити, що можливості Shopify дещо обмежені.

3. Залежність від додатків. Магазин додатків Shopify надає широкий спектр розширень та інтеграцій для розширення функціональності інтернет-магазину. Однак, значна залежність від сторонніх додатків може збільшити витрати, оскільки багато з них мають додаткову плату. Крім того, використання великої кількості додатків може потенційно вплинути на загальну продуктивність і швидкість завантаження вебсайту.

4. Shopify, як хостингова платформа, зберігає контроль над базовою інфраструктурою та серверами. Хоча вони забезпечують безпеку даних і резервне копіювання, бізнес має менший контроль над своїми даними порівняно з рішеннями, розміщеними на власному хостингу. Залежність від інфраструктури Shopify означає, що компанії покладаються на здатність платформи підтримувати конфіденційність і безпеку даних.

5. Обмежений контроль SEO. Shopify надає базові функції SEO та інструменти оптимізації, але рівень контролю та кастомізації може бути обмеженим у порівнянні з автономними системами управління контентом. Компанії можуть мати меншу гнучкість у впровадженні передових методів SEO або оптимізації певних елементів свого вебсайту для пошукових систем.

6. Проблеми міграції платформ. Якщо бізнес вирішить перейти на іншу платформу електронної комерції в майбутньому, міграція інтернет-магазину з Shopify може бути складним завданням. Хоча Shopify надає інструменти для експорту певних даних, можуть виникнути складнощі з перенесенням інформації про товари, налаштувань SEO та даних клієнтів на нову платформу. Процес міграції може вимагати технічних знань і ретельного планування для забезпечення плавного переходу. Був проведений комплексний аналіз інших платформ для розробки інтернет-магазинів (табл.1.1):

1. Horoshop – українська спеціалізована платформа для інтернет-магазинів [10].

2. WooCommerce – популярне рішення на основі CMS WordPress [11].

Таблиця 1.1 – Порівняння популярних платформ для створення інтернет-магазинів

| Критерій              | Платформи для порівняння                                   |                                                                         |                                                                |
|-----------------------|------------------------------------------------------------|-------------------------------------------------------------------------|----------------------------------------------------------------|
|                       | Shopify                                                    | Horoshop                                                                | WooCommerce                                                    |
| Тип платформи         | SaaS-платформа (хмарний сервіс)                            | Хмарна платформа для інтернет-магазинів                                 | Плагін електронної комерції для WordPress                      |
| Складність запуску    | Дуже проста — магазин можна запустити за кілька годин      | Простий запуск із готовими шаблонами                                    | Потребує встановлення WordPress, плагінів і хостингу           |
| Наявність хостингу    | Вбудований хостинг                                         | Вбудований хостинг                                                      | Потрібно окремо купувати хостинг                               |
| Кастомізація          | Обмежена можливостями тем і додатків                       | Налаштування через адмін-панель і шаблони                               | Дуже висока завдяки open-source та плагінам                    |
| Інтеграції та додатки | Понад 100 платіжних шлюзів та великий маркетплейс додатків | Інтеграції з українськими службами доставки, CRM і платіжними сервісами | Велика кількість плагінів та інтеграцій                        |
| SEO-можливості        | Вбудовані SEO-інструменти                                  | SEO-функції доступні у базових налаштуваннях                            | Потребує додаткових SEO-плагінів (наприклад Yoast)             |
| Платіжні системи      | Shopify Payments + понад 100 платіжних сервісів            | Інтеграції з українськими платіжними системами                          | Підтримує багато платіжних шлюзів через плагіни                |
| Масштабованість       | Висока — підходить для великих магазинів                   | Орієнтована на малий і середній бізнес                                  | Висока, але залежить від хостингу                              |
| Вартість              | від ~\$29/місяць                                           | від ~288 грн/місяць                                                     | Плагін безкоштовний, але потрібні витрати на хостинг і плагіни |

Порівняння показує, що платформи мають різні підходи до створення інтернет-магазинів. Платформа Shopify є повністю хмарним рішенням із готовою інфраструктурою та широкими можливостями масштабування. Українська платформа Horoshop орієнтована на локальний ринок і має інтеграції з українськими сервісами доставки та оплати. Натомість WooCommerce забезпечує найбільшу гнучкість і можливість індивідуальної розробки, але потребує технічного налаштування та підтримки. Хоча він пропонує варіанти перекладу для певних елементів, створення повністю локалізованого багатомовного магазину може вимагати використання сторонніх додатків або

індивідуальної розробки. Це може додати складнощів і витрат для компаній, які орієнтуються на кілька мовних ринків.

Нижче наведено демонстрацію вигляду робочого простору для створення потенційного сайту, який розробляється, використовуючи Shopify (рис. 1.1).

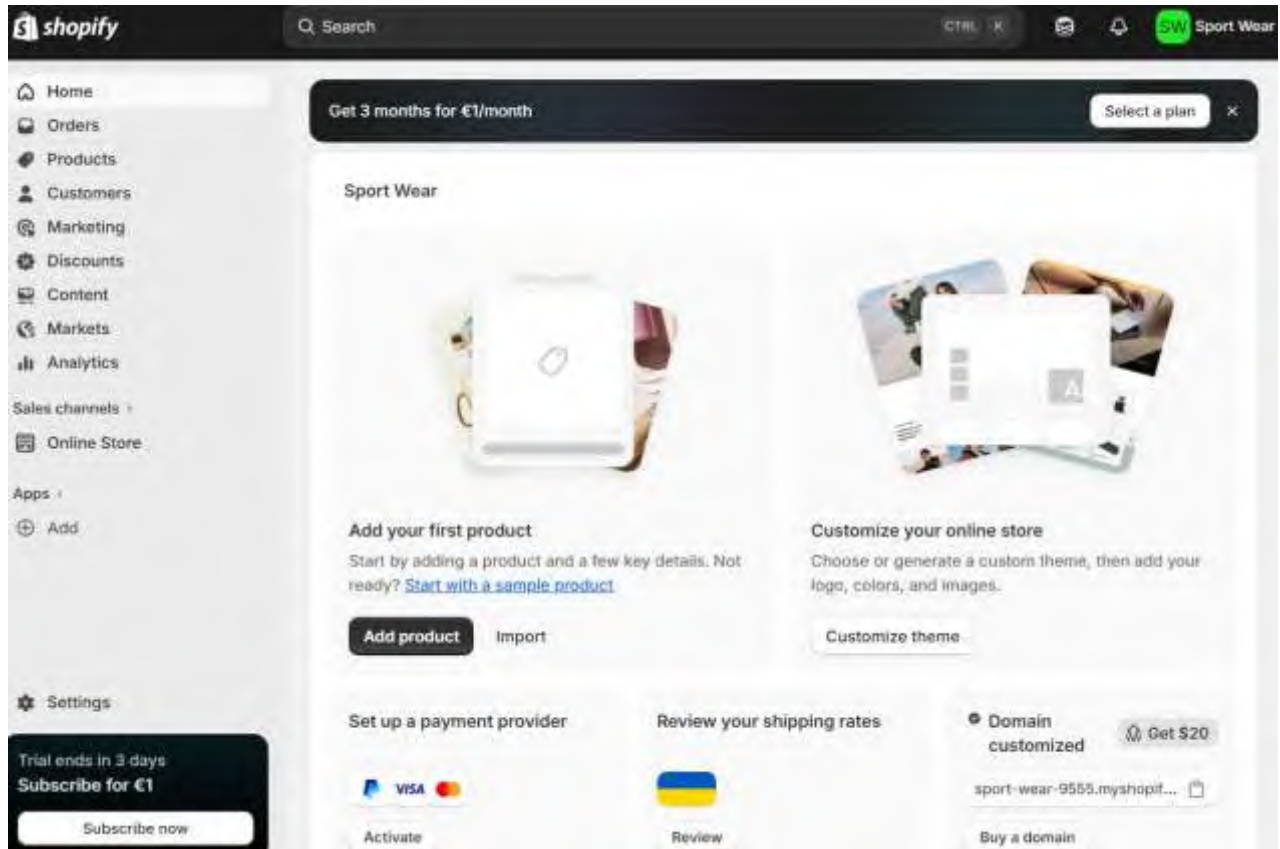


Рисунок 1.1 – Приклад стартового вікна для розробки інтернет-магазину на платформі Shopify [12]

Отже, Shopify пропонує низку переваг для бізнесу, який прагне створити та керувати своїми інтернет-магазинами. Зручний інтерфейс, широкий вибір тем, надійні функції для управління продуктами та виконання замовлень роблять його популярним вибором серед платформ електронної комерції. Крім того, наявність численних додатків та інтеграцій дозволяє компаніям налаштовувати свої інтернет-магазини відповідно до конкретних потреб.

На основі аналізу табл. 1.1 та власного експерименту можна порекомендувати розробникам використати вітчизняну платформу, переваги якої представлені на головній сторінці сайту (рис. 1.2).



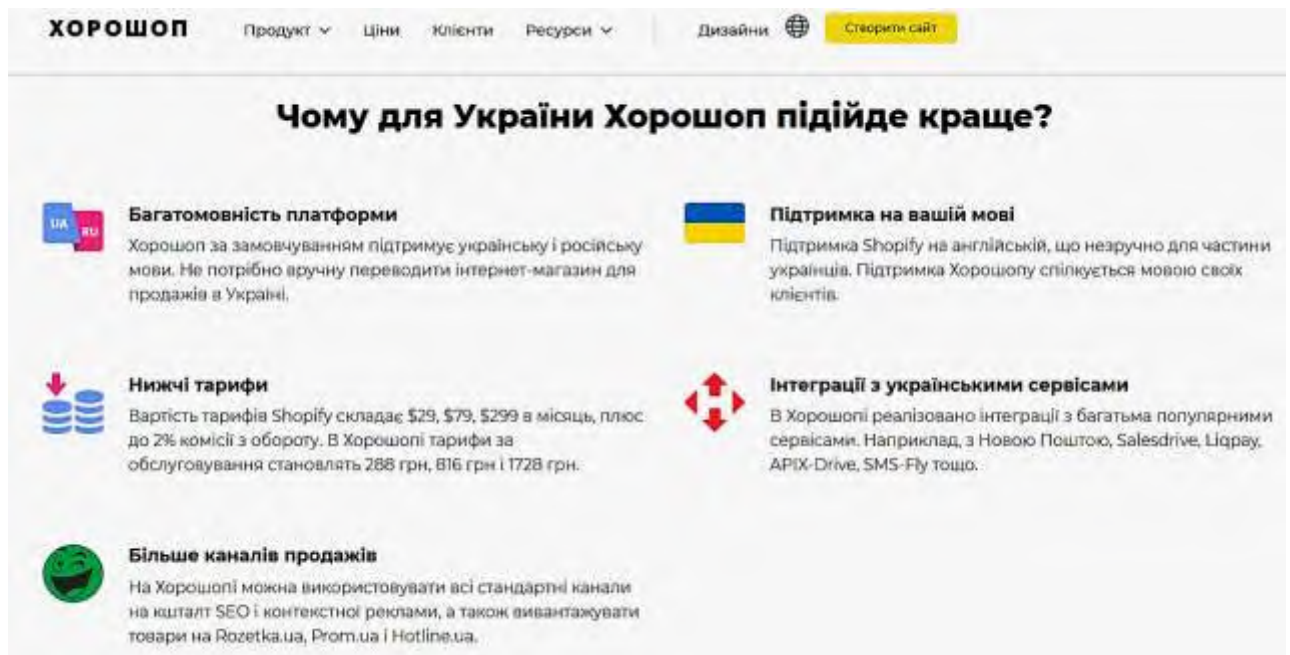


Рисунок 1.2 – Переваги використання української платформи «Хорошоп»

Отже, важливо при виборі враховувати і потенційні недоліки використання різних платформ, альтернативних Shopify. Структура витрат, обмеження кастомізації, залежність від додатків та обмежений контроль над даними і SEO – це фактори, які компанії повинні зважити на переваги платформи. Крім того, при розгляді платформи як довгострокового рішення слід враховувати проблеми, пов’язані з міграцією платформи та обмеженими багатомовними можливостями.

### 1.3 Аналіз конкурентів на ринку інтернет-магазинів спортивного одягу

При створенні власного інтернет-магазину для порівняння були детально проаналізовані офіційні вебсайти таких популярних брендів спортивного одягу, як Adidas [13], PUMA [14], Skechers [15]. Фрагмент головної сторінки сайту бренду Adidas наведено на рис. 1.3.

Після вибору був проведений аналіз інтернет-магазинів вебсайтів та те, що подобається і чого не вистачає користувачам. За результатами аналізу спочатку детально викладені переваги інтернет-магазину бренду Adidas, враховуючи критерії зручності з боку користувача. Результати наведені у табл. 1.2.

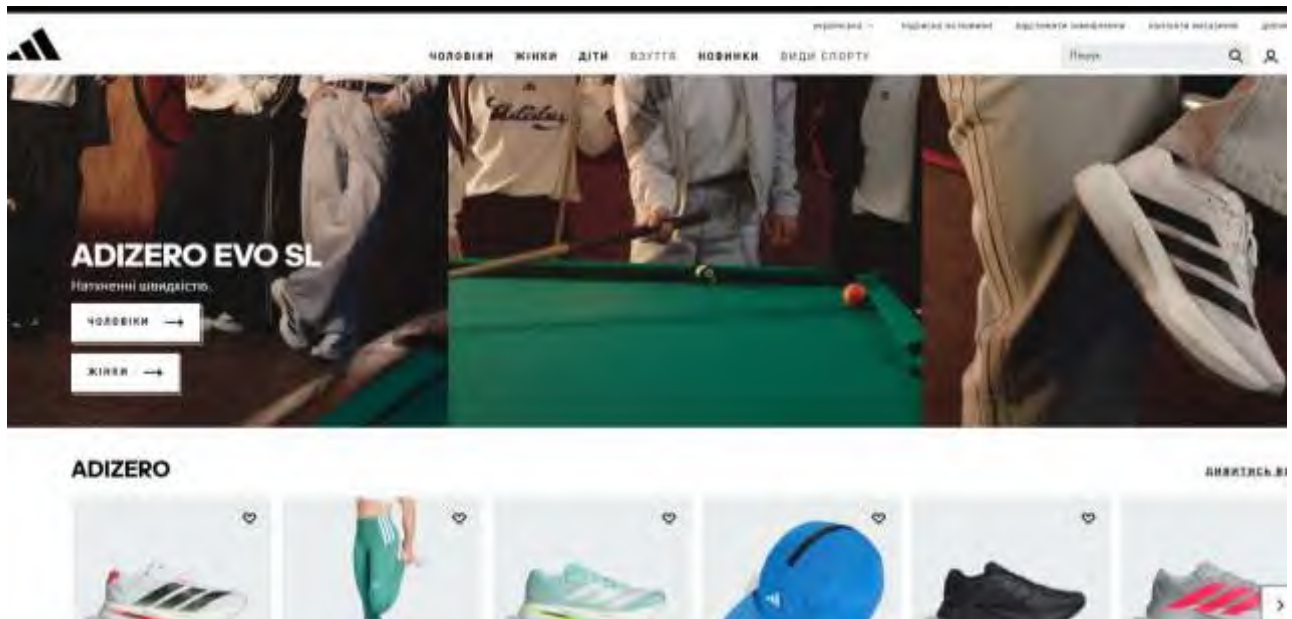


Рисунок 1.3 – Вигляд головної сторінки інтернет-магазину Adidas [13]

Таблиця 1.2 – Аналіз сильних сторін інтернет-магазину Adidas

| Критерій                | Характеристика вебзастосунку за критерієм                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Зовнішній вигляд        | Вебсайт Adidas має візуально привабливий дизайн, що відповідає фірмовому стилю бренду, включає в себе впізнаваний логотип, типографіку та кольорову гаму бренду (чорний, м'ятний, білий), створюючи послідовний і цілісний візуальний досвід. Використання високоякісних зображень товарів і привабливих візуальних ефектів покращує загальну естетику магазину. Макет чистий і організований, що полегшує покупцям навігацію та пошук потрібних товарів. Включення зображень стилю життя та відео за участю спортсменів і впливових людей додає привабливості, що спонукає до прагнення до успіху, або приєднання через володіння товаром. |
| Навігація               | Магазин має чітке та інтуїтивно зрозуміле навігаційне меню, яке класифікує товари за статтю, видом спорту та колекцією. Це допомагає покупцям швидко знайти потрібний розділ і знайти конкретні товари.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Функціональність пошуку | Adidas має пошуковий рядок, який дозволяє покупцям шукати товари за назвою, категорією або ключовими словами. Результати пошуку зазвичай релевантні, що дозволяє покупцям ефективно знаходити конкретні товари.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Презентація продукту    | Кожна сторінка товару в онлайн-магазині Adidas містить вичерпну інформацію, включно з деталями продукту, розмірними таблицями, відгуками клієнтів та супутніми товарами. Включення декількох зображень, що демонструють товар під різними кутами та крупним планом, допомагає у процесі прийняття рішення.                                                                                                                                                                                                                                                                                                                                  |
| Опції кастомізації      | Adidas пропонує опції кастомізації для певних продуктів, таких як персоналізовані кросівки чи футболки. Ця функція додає рівня персоналізації та унікальності до досвіду покупок.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Адаптивний дизайн       | Інтернет-магазин розроблений таким чином, щоб бути адаптивним, підлаштовуючись під різні розміри екранів і пристроїв. Це забезпечує послідовний і приємний досвід для клієнтів, які заходять на сайт зі стаціонарних комп'ютерів, планшетів або мобільних пристроїв.                                                                                                                                                                                                                                                                                                                                                                        |

### Продовження таблиці 1.2

|                              |                                                                                                                                                                                                                                                                                  |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Процес оформлення замовлення | Хоча процес оформлення замовлення, як правило, простий, його можна вдосконалити, зменшивши кількість необхідних кроків і забезпечивши безперебійний процес оплати. Оптимізація процесу та пропозиція декількох варіантів оплати може підвищити зручність використання.           |
| Доступність                  | Щоб задовольнити різноманітну базу користувачів, забезпечення таких функцій доступності, як альтернативний текст для зображень, належний колірний контраст і клавіатурна навігація, може покращити зручність використання інтернет-магазину для людей з обмеженими можливостями. |

Отже, інтернет-магазин Adidas пропонує привабливий візуальний дизайн, який відповідає ідентичності бренду. Навігаційне меню та функція пошуку полегшують клієнтам пошук товарів, а всебічна презентація продукції допомагає приймати обґрунтовані рішення. Можливість кастомізації додає індивідуального підходу, а адаптивний дизайн забезпечує однаковий досвід на всіх пристроях. Хоча процес оформлення замовлення та функції доступності можна вдосконалити, загалом інтернет-магазин Adidas забезпечує візуально приємну та зручну платформу для клієнтів, щоб вони могли досліджувати та купувати бажані товари.

Перейдемо до аналізу переваг за подібними критеріями інтернет-магазину PUMA (рис. 1.4) і Skechers [15]. Аналітичну інформацію об'єднано в табл. 1.3.

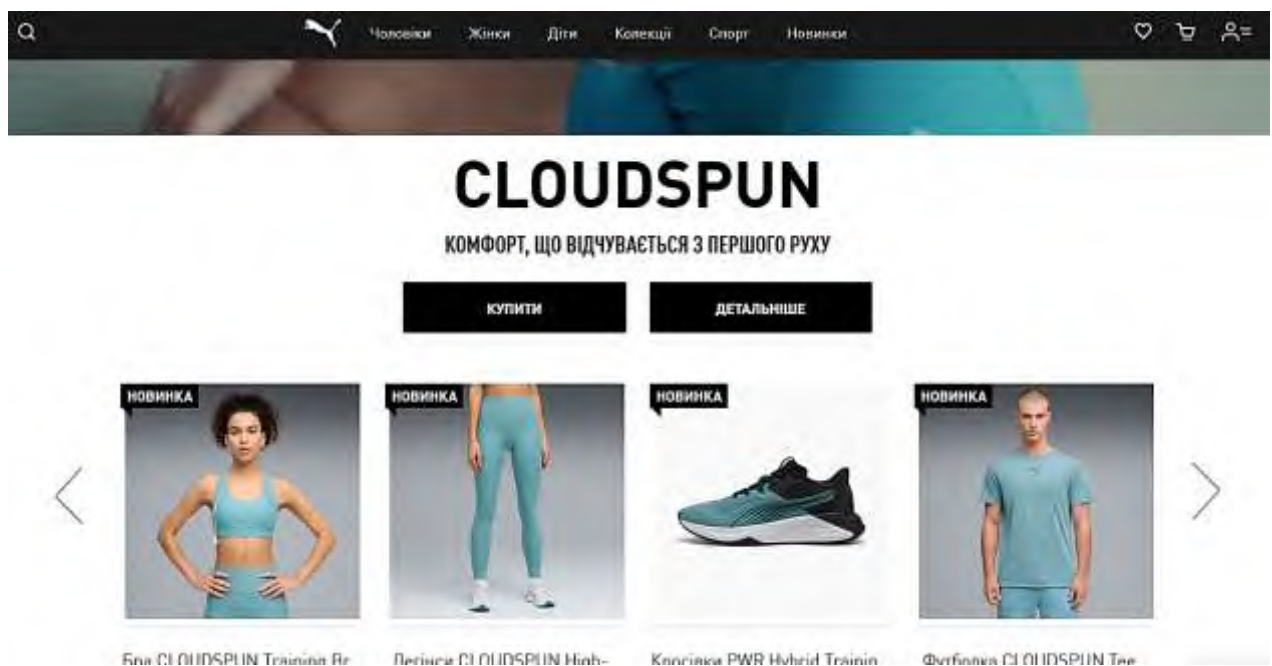


Рисунок 1.4 – Дизайн головної сторінки інтернет-магазину PUMA [14]

Таблиця 1.3 – Порівняльний аналіз інтернет-магазинів спортивного одягу

| Критерій                                | PUMA                                                                                                                       | Skechers                                                                                                           |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Зовнішній вигляд інтерфейсу користувача | Сучасний мінімалістичний дизайн, використання великих банерів, брендова кольорова гама, професійні фото продукції          | Простий та функціональний дизайн із фокусом на каталог товарів та акції                                            |
| Зручність навігації                     | Чітка структура каталогу з поділом на чоловічі, жіночі та дитячі товари, а також на види активності (біг, тренування тощо) | Каталог структурований за категоріями (чоловіче, жіноче, дитяче взуття), що дозволяє швидко знайти потрібний товар |
| Презентація продукту                    | Детальні сторінки товару: якісні зображення, опис характеристик, розміри, рекомендації                                     | Фото товарів, опис моделей і технологій комфорту (Air-Cooled Memory Foam, Arch Fit)                                |
| Персоналізація замовлення               | Можливість вибору розміру, кольору, перегляду схожих товарів, рекомендацій                                                 | Вибір розміру, швидке додавання товару до кошика, інколи пропозиції супутніх товарів                               |
| Адаптивний дизайн                       | Сайт оптимізований для мобільних пристроїв, що дозволяє зручно здійснювати покупки зі смартфонів                           | Адаптивна версія для мобільних пристроїв із збереженням основних функцій каталогу                                  |
| Процес оформлення замовлення            | Класичний багатокроковий checkout: кошик → контактні дані → доставка → оплата                                              | Спрощене оформлення замовлення з можливістю оплати при отриманні                                                   |

Аналіз інтернет-магазинів показує, що обидва вебресурси реалізують ключові функції електронної комерції: каталог товарів, систему навігації, презентацію продукції та процес оформлення замовлення. Інтернет-магазин бренду PUMA має більш сучасний дизайн та складнішу структуру інтерфейсу, орієнтовану на міжнародний бренд і широкий асортимент спортивного одягу. Водночас сайт Skechers характеризується простішим інтерфейсом і спрощеним процесом покупки, що може бути зручним для користувачів, які віддають перевагу швидкому оформленню замовлення.

Додатковий аналіз проведено для оцінювання якості інтерфейсу вебсайтів інтернет-магазинів спортивного одягу. Оцінювання виконано за 5-бальною шкалою, де 1 – дуже низький рівень, 5 – високий рівень). Критерії відповідають основним елементам usability та функціональності e-commerce вебсайтів. Інтегральна оцінка представлена в табл. 1.4.

Таблиця 1.4 – Оцінка якості вебінтерфейсу інтернет-магазинів спортивного одягу

| Критерій                                 | Adidas | PUMA | Skechers |
|------------------------------------------|--------|------|----------|
| Зовнішній вигляд інтерфейсу користувача  | 5      | 5    | 4        |
| Зручність навігації                      | 5      | 4    | 4        |
| Презентація продукту                     | 5      | 4    | 4        |
| Персоналізація замовлення                | 4      | 4    | 3        |
| Адаптивний дизайн                        | 5      | 5    | 4        |
| Процес оформлення замовлення             | 4      | 4    | 4        |
| Система пошуку та фільтрації товарів     | 5      | 4    | 3        |
| Інформаційна підтримка користувача       | 4      | 4    | 3        |
| Загальна середня оцінка функціональності | 4,6    | 4,3  | 3,6      |

Проведений аналіз показує, що всі три інтернет-магазини мають функціонал, характерний для сучасних платформ електронної комерції. Сайт бренду Adidas демонструє найвищий рівень якості інтерфейсу та презентації продукції, що пояснюється значними інвестиціями у цифрову інфраструктуру глобального бренду. Adidas пропонує візуально привабливий дизайн, який відповідає корпоративному стилю, що разом з високоякісними зображеннями товарів і привабливими візуальними ефектами покращує естетику магазину. Навігація інтернет-магазину є чіткою та інтуїтивно зрозумілою. Правильніше розташована фільтрація та простота вибору товарів за статтю. Це дозволяє мені легко знайти конкретний фільтр, який мене цікавить. Adidas пропонує опції кастомізації для певних продуктів, що дозволяє персоналізувати досвід покупок.

Інтернет-магазин PUMA має сучасний дизайн, адаптивний інтерфейс та розвинену структуру каталогу, що забезпечує зручну навігацію і швидкий доступ до категорій товарів. Каталог поділяється за призначенням продукції (біг, тренування, lifestyle), що полегшує пошук товарів для різних видів активності.

Сайт Skechers має більш просту структуру та інтерфейс, однак забезпечує основні функції електронної комерції: каталог товарів, вибір розміру та оформлення замовлення.

Отже, проведене порівняння дозволяє визначити ключові функціональні елементи, які доцільно врахувати під час розробки власного інтернет-магазину спортивного одягу.

## РОЗДІЛ 2

### ФОРМУВАННЯ КОНЦЕПЦІЇ ТА ОБГРУНТУВАННЯ МЕТОДІВ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ СПОРТИВНОГО ОДЯГУ

#### 2.1 Складання технічних характеристик функціональних вимог інтернет-магазину спортивного одягу. Визначення цільової аудиторії

На основі аналізу функціональних можливостей сучасних інтернет-магазинів спортивного одягу можна визначити ключові вимоги до розроблюваного вебресурсу. Робота над власним проектом розпочинається зі складання консолідованого пулу функціональних вимог до інтернет-магазину спортивного одягу (табл. 2.1) з урахуванням того, що деякі продавці, які не є прямими дилерами однієї марки, створюють мультибрендовий маркет-плейс, де можна одразу пропонувати різні брендові товари, поєднуючи сильні сторони різних виробників та створювати власну систему маркетингу, задовольняючи індивідуальні потреби користувачів.

Таблиця 2.1 – Функціональні вимоги до інтернет-магазину спортивного одягу

| Функціональна вимога   | Опис функції                                                                                         | Мета реалізації                                             |
|------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| Каталог товарів        | Відображення товарів за категоріями (чоловічий, жіночий, дитячий одяг, взуття, аксесуари)            | Забезпечення зручного доступу користувачів до асортименту   |
| Пошук товарів          | Наявність пошукового рядка з можливістю швидкого знаходження товарів за назвою або ключовими словами | Спрощення процесу пошуку продукції                          |
| Фільтрація товарів     | Можливість сортування товарів за ціною, розміром, брендом, типом продукції                           | Полегшення вибору потрібного товару                         |
| Сторінка товару        | Детальна інформація про товар: фото, опис, характеристики, доступні розміри та кольори               | Надання повної інформації для прийняття рішення про покупку |
| Кошик покупця          | Можливість додавання товарів до кошика, зміни кількості та видалення товарів                         | Формування замовлення перед оплатою                         |
| Оформлення замовлення  | Введення контактних даних, вибір способу доставки та оплати                                          | Реалізація процесу покупки                                  |
| Реєстрація користувача | Створення особистого кабінету покупця                                                                | Збереження історії замовлень та персоналізація сервісу      |

*Продовження таблиці 2.1*

|                             |                                                                           |                                             |
|-----------------------------|---------------------------------------------------------------------------|---------------------------------------------|
| Адаптивний дизайн           | Коректне відображення сайту на різних пристроях (ПК, планшети, смартфони) | Забезпечення зручності використання сайту   |
| Інтеграція платіжних систем | Підтримка онлайн-оплати (банківські картки, електронні платіжні системи)  | Забезпечення безпечних фінансових операцій  |
| Інтеграція служб доставки   | Підключення служб доставки для автоматизації відправлення товарів         | Оптимізація логістики                       |
| Система відгуків            | Можливість залишати відгуки та оцінки товарів                             | Підвищення довіри покупців                  |
| Адміністративна панель      | Інструменти управління товарами, замовленнями та користувачами            | Ефективне адміністрування інтернет-магазину |

Отже, основними рисами кінцевого продукту у вигляді вебінтерфейсу інтернет-магазину є наявність зручного каталогу товарів, ефективної системи пошуку та фільтрації, адаптивного дизайну, а також інтеграція платіжних систем і служб доставки. Реалізація зазначених функцій забезпечить зручність користування сайтом, підвищить ефективність процесу покупки та сприятиме покращенню користувацького досвіду і ефективності роботи інтернет-магазину [1, 2].

Структура та асортимент магазину спортивного одягу можуть варіюватися залежно від різних факторів, таких як розмір магазину, його місцезнаходження та цільова аудиторія. Нижче наведено приклад створення структури інтернет-магазину (рис. 2.1).



Рисунок 2.1 – Схема архітектури повного сайту інтернет-магазину



При технічному плануванні розробки необхідно продумати й суто технічні вимоги до якісних характеристик інтернет-магазину, як вебдодатку. Найбільш суттєві аспекти наведені в табл. 2.2.

Таблиця 2.2 – Вимоги до якісних характеристик інтернет-магазину

| Показник нефункціональної вимоги   | Опис вимоги                                                                 | Мета реалізації                                                     |
|------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------|
| Продуктивність                     | Час завантаження сторінок не більше 2–3 секунд                              | Забезпечення швидкої роботи сайту та зменшення відтоку користувачів |
| Масштабованість                    | Здатність системи обробляти збільшення кількості користувачів і замовлень   | Підтримка розвитку бізнесу та зростання навантаження                |
| Безпека                            | Використання HTTPS, захист персональних даних, безпечні платіжні транзакції | Захист даних користувачів і фінансових операцій                     |
| Надійність                         | Стабільна робота сайту без збоїв і помилок                                  | Забезпечення безперервного доступу до сервісу                       |
| Доступність (Availability)         | Доступність сайту 24/7                                                      | Можливість здійснення покупок у будь-який час                       |
| Адаптивність                       | Коректна робота на різних пристроях і браузерах                             | Покращення користувацького досвіду                                  |
| Юзабіліті (зручність використання) | Інтуїтивно зрозумілий інтерфейс                                             | Спрощення взаємодії користувача із сайтом                           |
| SEO-оптимізація                    | Оптимізація структури сайту для пошукових систем                            | Підвищення видимості сайту в пошукових системах                     |
| Сумісність (Compatibility)         | Підтримка різних браузерів (Chrome, Firefox, Safari тощо)                   | Забезпечення коректної роботи для всіх користувачів                 |
| Підтримка та супровід              | Можливість оновлення системи, виправлення помилок, додавання нових функцій  | Забезпечення довготривалої експлуатації системи                     |
| Локалізація                        | Підтримка кількох мов (наприклад, українська та англійська)                 | Розширення аудиторії користувачів                                   |
| Відмовостійкість                   | Здатність системи відновлювати роботу після збоїв                           | Мінімізація втрат даних і простоїв                                  |

Нефункціональні вимоги визначають якісні характеристики інтернет-магазину та є критично важливими для його успішного функціонування. До основних із них належать продуктивність, безпека, надійність, доступність, адаптивність і зручність використання. Особливу увагу слід приділити захисту персональних даних і стабільності роботи системи, оскільки це безпосередньо



впливає на довіру користувачів. Крім того, SEO-оптимізація та сумісність із різними пристроями забезпечують розширення аудиторії та покращення позицій у пошукових системах. Врахування нефункціональних вимог на етапі проєктування дозволяє створити конкурентоспроможний і надійний вебресурс [2, 3]. Опишемо покроковий процес планування і обґрунтування структури.

1. Планування і структура. Магазин повинен мати добре організоване планування, яке дозволяє покупцям легко орієнтуватися в різних секціях. Магазин поділений на окремі зони для різних типів одягу, щоб покупцям було зручно знаходити те, що вони шукають. Зони викладки повинні бути візуально привабливими і висвітлювати найновіші або найпопулярніші товари. Візуалізація принципів архітектури відштовхується від запланованої мапи сайту (рис. 2.1).

2. Фільтри та сортування. Магазин пропонує різні фільтри, такі як розмір, кольори, бренд, сортування ціни тощо. Крім того, функції сортування, такі як сортування за популярністю, новинками або ціною, можуть бути доступні, щоб покупці могли швидко знайти потрібні товари.

3. Продуманий і широкий асортимент спортивних товарів для різних видів спорту та активностей, включаючи одяг, взуття та аксесуари. Товари від популярних і авторитетних брендів, які відомі своєю якістю та продуктивністю., а також можливість пропонувати різноманітні стилі та дизайн, щоб задовольнити різні вподобання клієнтів, варіанти для різних вікових груп, статей і розмірів, щоб задовольнити потреби різноманітної клієнтської бази.

4. Одяг, взуття та аксесуари. Наявність різноманітного спортивного одягу, взуття та аксесуарів, такого як футболки, шорти, куртки, кросівки, кеди, сумки, рюкзаки та головні убори, що підходить для різних видів діяльності та кліматичних умов.

5. Визначення цільової аудиторії. Щоб точніше визначити цільову аудиторію, важливо враховувати такі фактори, як демографічні показники та вподобання клієнтів. Цільова аудиторія включає в себе дітей, молодь та дорослих людей віком до 50 років, адже це охоплює широкий демографічний діапазон, що

може значно розширити ринковий потенціал інтернет-магазину спортивного одягу. Орієнтуючись на різні вікові групи, магазин може залучити більшу клієнтську базу і потенційно збільшити продажі.

Для досягнення поставленої мети необхідно вирішити такі основні завдання:

- реалізувати зручний та інтуїтивно зрозумілий користувацький інтерфейс;
- розробити функціональний каталог товарів із можливістю пошуку та фільтрації;
- забезпечити відображення детальної інформації про товар;
- реалізувати механізм кошика та оформлення замовлення;
- забезпечити адаптивність вебзастосунку для різних типів пристроїв;
- організувати збереження та обробку даних користувачів і товарів;
- інтегрувати сервіси для роботи з медіаконтентом.

Для реалізації зазначених завдань було обрано сучасний стек технологій, що відповідає вимогам до продуктивності, масштабованості та зручності розробки.

## **2.2 Підбір та характеристика технологій для розробки інтернет-магазину**

У ході виконання кваліфікаційної роботи для проєкту були обрані такі технології: React для створення користувацького інтерфейсу, Redux для управління станом застосунку, Firebase для обробки бази даних, Cloudinary. Як середовище розробки обрано Visual Studio Code, що є популярним редактором коду з широкою підтримкою розширень, інтеграцією з системами контролю версій та зручними інструментами для роботи з JavaScript і React. Надамо коротке пояснення щодо вибору.

Visual Studio Code (VS Code) – популярний редактор вихідного коду, розроблений компанією Microsoft. Він відрізняється легкістю, широкими

можливостями налаштування та універсальністю, що робить його кращим вибором серед розробників, які використовують різні мови програмування та платформи [16].

Ключові можливості та характеристики Visual Studio Code:

1. Інтерфейс користувача. VS Code має чистий та інтуїтивно зрозумілий користувацький інтерфейс з бічною панеллю для навігації, центральною областю редагування та рядком стану для відображення інформації та інструментів. Він забезпечує вільний робочий простір, дозволяючи розробникам зосередитися на своєму коді (рис. 2.2 – 2.3).

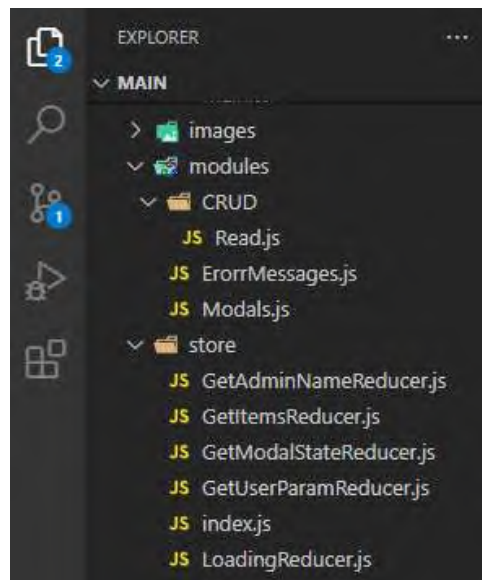


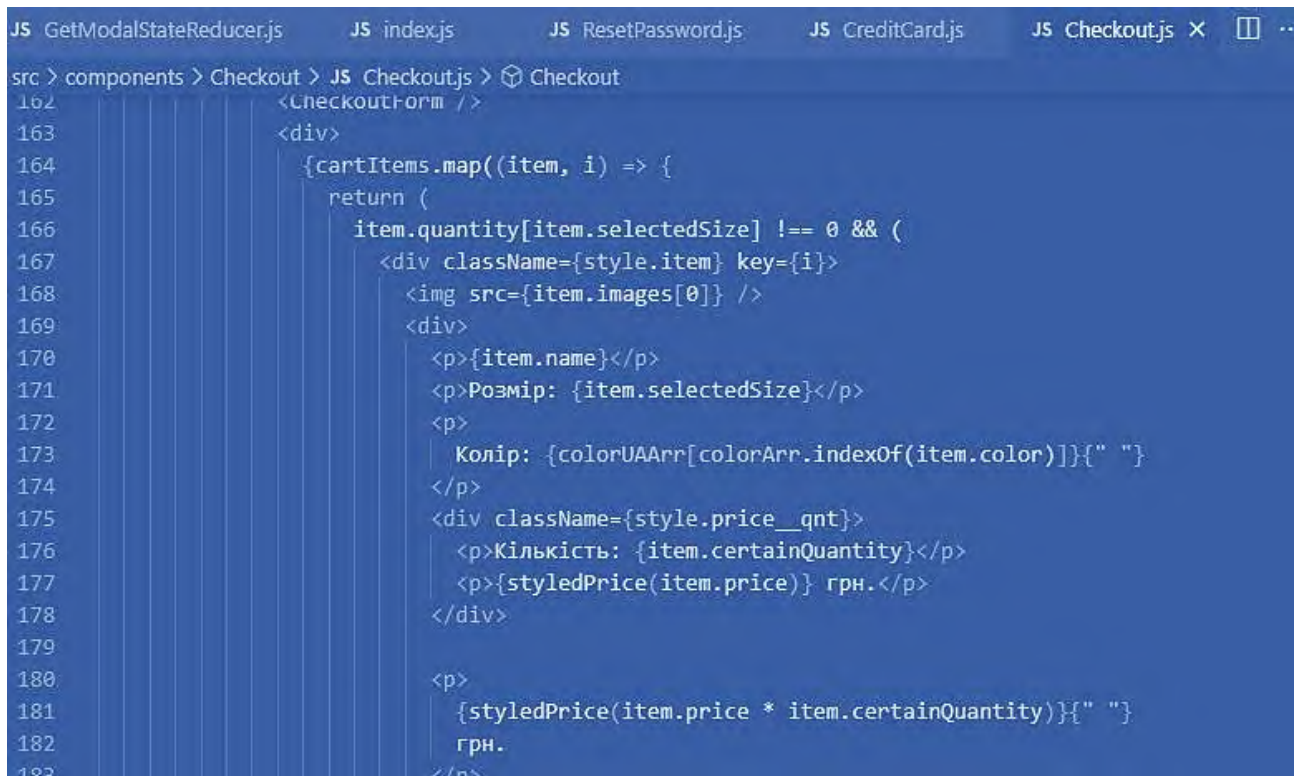
Рисунок 2.2 – Вигляд бічної панелі Visual Studio Code

Бічна панель Visual Studio Code забезпечує швидкий доступ до різних функцій та можливостей редактора коду. Бічна панель містить іконки навігації, такі як Провідник, Розширення та інші, що дозволяють користувачам легко перемикаватися між різними файлами та інструментами в редакторі.

2. Крос-платформна сумісність. Visual Studio Code доступний для Windows, macOS та Linux, що забезпечує сумісність та однакову роботу з різними операційними системами.

3. Розширення та маркетплейс. Однією з визначних особливостей VS Code є його широка екосистема розширень. Розробники можуть налаштовувати

та покращувати свій досвід кодування, встановлюючи широкий спектр розширень для різних мов програмування, фреймворків та інструментів. На Visual Studio Code Marketplace розміщено величезну колекцію розширень, створених спільнотою.



```

JS GetModalStateReducer.js JS index.js JS ResetPassword.js JS CreditCard.js JS Checkout.js X ..
src > components > Checkout > JS Checkout.js > Checkout
162 <CheckoutForm />
163 <div>
164   {cartItems.map((item, i) => {
165     return (
166       item.quantity[item.selectedSize] !== 0 && (
167         <div className={style.item} key={i}>
168           <img src={item.images[0]} />
169           <div>
170             <p>{item.name}</p>
171             <p>Розмір: {item.selectedSize}</p>
172             <p>
173               Колір: {colorUArr[colorArr.indexOf(item.color)]}{" "}
174             </p>
175             <div className={style.price_qnt}>
176               <p>Кількість: {item.certainQuantity}</p>
177               <p>{styledPrice(item.price)} грн.</p>
178             </div>
179           </div>
180           <p>
181             {styledPrice(item.price * item.certainQuantity)}{" "}
182             грн.
183           </p>

```

Рисунок 2.3 – Вигляд центральної області Visual Studio Code

4. Наявність мовної підтримки для багатьох мов програмування, включаючи такі популярні, як JavaScript, Python, C++, Java, HTML, CSS та багато інших. Ця опція значно полегшує процес написання коду та зменшує кількість помилок завдяки підсвічуванню синтаксису, завершення коду, форматування, розбиття на рядки та можливості налагодження з урахуванням специфіки для кожної мови програмування.

5. Інтегрований термінал. Дозволяє розробникам запускати команди, компілювати код і виконувати різні завдання без перемикання на окремий термінальний додаток. Це полегшує безперебійний робочий процес розробки в самому редакторі (рис. 2.4).

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/invertible-env-367910/overview
Hosting URL: https://invertible-env-367910.web.app
PS D:\diplom\main>
* History restored

warning: in the working copy of '.firebase', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'firebase.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'public/index.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of '.firebase/hosting.YnVpbGQ.cache', LF will be replaced by CRLF the next time Git touches it
PS D:\diplom\main>

```

Рисунок 2.4 – Вигляд інтегрованого терміналу Visual Studio Code

6. Інтеграція з Git. VS Code має вбудовану інтеграцію з Git, що дозволяє контролювати версії та співпрацювати з колегами по команді (рис. 2.5). Розробники можуть легко переглядати та керувати репозиторіями Git'а, фіксувати зміни, переглядати різниці та виконувати операції з Git'ом, не виходячи з редактора [17].

```

- undefinedVariable = {};
- undefinedVariable.prop = 5;
+ var definedVariable = {};
+ definedVariable.prop = 5;

```

Рисунок 2.5 – Вигляд різних версій проекту Visual Studio Code

7. Можливості налагодження. Visual Studio Code забезпечує надійну підтримку налагодження для різних мов і платформ. Вона пропонує покрокове налагодження, перевірку змінних та інші важливі функції налагодження, які допомагають розробникам виявляти та виправляти проблеми в коді.

8. IntelliSense. Інтелектуальна функція завершення коду у VS Code, яка надає контекстно-залежні пропозиції під час введення коду (рис. 2.6). Вона пропонує автозавершення, підказки щодо підписів функцій та документацію, щоб пришвидшити кодування та зменшити кількість помилок [18].

9. Автоматизація завдань. VS Code підтримує автоматизацію завдань за допомогою інтегрованого бігуна завдань. Розробники можуть визначати власні

завдання збірки, запускати скрипти і виконувати різні автоматизовані завдання в редакторі, підвищуючи продуктивність і ефективність робочого процесу.

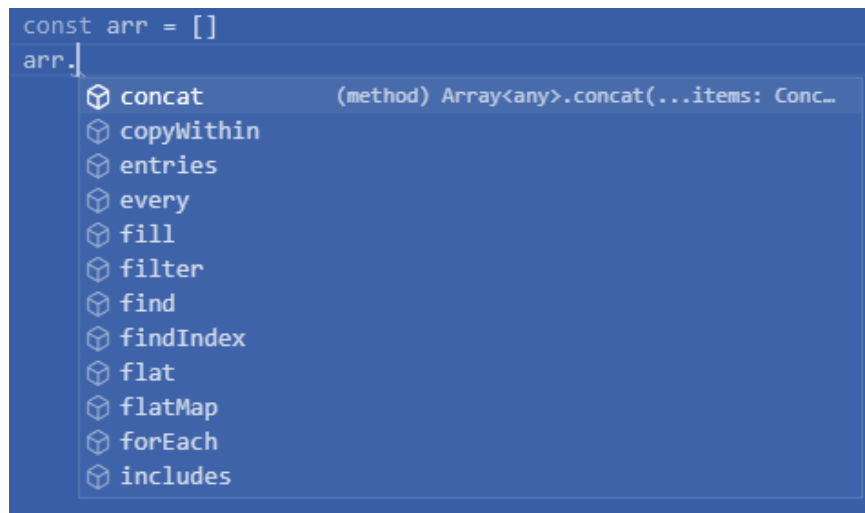


Рисунок 2.6 – Вигляд IntelliSense Visual Studio Code

10. Кастомізація. Visual Studio Code дуже добре налаштовується, що дозволяє розробникам персоналізувати своє середовище кодування. Користувачі можуть змінювати теми, встановлювати розширення, налаштовувати параметри та створювати власні комбінації клавіш, щоб пристосувати редактор до своїх уподобань [19].

Загалом, Visual Studio Code надає потужне та гнучке середовище кодування з багатим набором функцій, розширень та можливостей налаштування, що робить його популярним вибором для розробників усіх рівнів. Його легкість, широка мовна підтримка та динамічна екосистема сприяють його широкому розповсюдженню у спільноті розробників.

У якості основного інструменту для створення користувацького інтерфейсу використано React. React – це потужна бібліотека JavaScript, яка використовується для створення динамічних користувацьких інтерфейсів та досвіду. Розроблена та підтримувана Facebook з березня 2013 року React широко використовується такими великими компаніями, як Instagram, Uber, Netflix, Twitter та багатьма іншими. Розробники цінують його за здатність створювати великі, швидкі та масштабовані вебдодатки.

«Одна з чудових частин React – це те, як з’являються додатки в міру їхнього створення. Якщо описувати цей процес у вигляді кроків, то отримаємо такий шлях [20]:

1. Декомпозиція UI у вигляді компонентів;
2. Відтворення даних без інтерактивності;
3. Визначення мінімальних станів для завдання інтерактивності;
4. Вибір компонентів, що володіють і змінюють стан;
5. Визначення зворотного потоку даних.

В основу React покладені компоненти, елементи, Virtual DOM і JSX.

Елементи – це об’єкти JavaScript, які представляють HTML-елементи (описують DOM-елементи). Вони є будівельними блоками React.

JSX – це препроцесор, що спрощує створення елементів і компонентів React. З ним потрібно набагато менше зусиль порівняно з класичним Java Script під час написання коду. JSX трансформується в JavaScript перед запуском у браузері. Він не є обов’язковим під час використання React. Робота з React без JSX комфортна і зручна, особливо якщо немає потреби налаштовувати компіляцію у своєму середовищі збірки.

Кожен елемент JSX є просто синтаксичним цукром для виклику "React.createElement (component, ...)". Тому все, що можна зробити за допомогою JSX, також можна зробити і за допомогою звичайного JavaScript.

Virtual DOM – представляє легковажну копію звичайного DOM. І відмінною особливістю React є те, що ця бібліотека працює саме з віртуальним DOM, а не звичайним [21].

Компоненти в React дозволяють розділити користувацький інтерфейс на самодостатні частини, які можна використовувати незалежно. Подібно до функцій JavaScript, компоненти приймають користувацькі дані (так звані пропси) і генерують React-елементи, які визначають візуальне представлення на екрані. Кожен компонент має властивості та стан, що забезпечує додаткову гнучкість та інтерактивність у додатку.

React надає ефективний спосіб керування станом елементів, використовуючи віртуальний DOM, що допомагає підвищити продуктивність. Завдяки декларативному API React розробники можуть зосередитися на описі бажаного результату, не турбуючись про зміни, що відбуваються в основі. Такий декларативний підхід спрощує розробку додатків, полегшуючи написання коду.

Однією з сильних сторін React є його здатність надавати детальний контроль над розміром додатку. Розробники можуть вибірково включати лише необхідні компоненти, забезпечуючи гнучкість при переході від односторонніх додатків (SPA) до мікросервісів. Це дозволяє повторно використовувати існуючі частини програми, що підвищує ефективність та масштабованість коду.

Здатність React забезпечувати детальний контроль над структурою та розміром застосунку пояснюється його компонентною архітектурою та модульним підходом. Розробник може включати лише необхідні елементи системи, що зменшує обсяг коду та підвищує продуктивність. Повторне використання (перевикористання) коду і можливість поступового розширення архітектури забезпечують гнучкість, ефективність та масштабованість вебзастосунку. Частини коду можна переносити між різними проектами, комбінувати у складніші інтерфейси. Наприклад, при проектуванні інтерфейсу один компонент “ProductCard” використовується у каталозі, у рекомендаціях, у кошику. Код, який був написаний під час роботи над сайтом, може бути повторно використаний для розробки мобільного застосунку, тобто немає необхідності наймати дві великі команди розробників [22].

Для управління станом застосунку використано Redux, що забезпечує централізоване зберігання даних і передбачуваність їх змін. Це особливо важливо для інтернет-магазину, де необхідно керувати станом кошика, списком товарів та даними користувача. Redux – це бібліотека управління передбачуваними станами для додатків на JavaScript, яка зазвичай використовується з такими фреймворками, як React та Angular. Вона забезпечує централізований підхід до управління станом додатку, що полегшує розуміння, тестування та підтримку складної логіки, пов’язаної зі станами [23].



Основна концепція Redux обертається навколо ідеї єдиного незмінного дерева станів. Весь стан програми зберігається в одному JavaScript-об'єкті, відомому як "сховище". Сховище відповідає за зберігання поточного стану і відправку дій для його зміни (рис. 2.7).

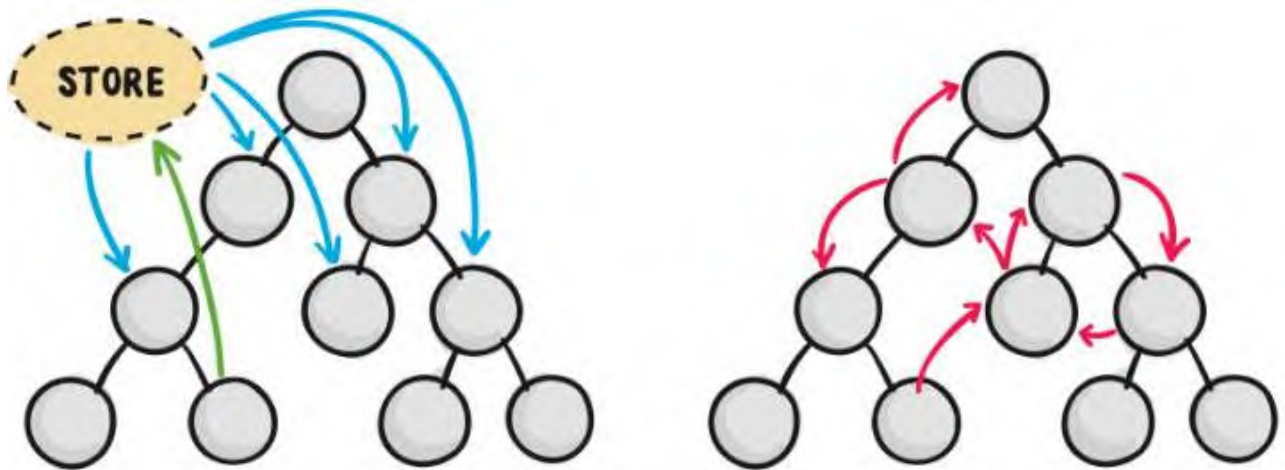


Рисунок 2.7 – Схематичне представлення, як працює Redux [23]

Стан в Redux не змінюється безпосередньо. Натомість зміни вносяться шляхом диспетчеризації дій, які є звичайними JavaScript-об'єктами з властивістю типу, що описує дію, яка виконується. Дії зазвичай викликаються взаємодією користувача або асинхронними операціями.

Редуктори відіграють важливу роль у Redux. Редуктор - це чиста функція, яка отримує на вхід поточний стан і дію, а на виході повертає новий стан на основі типу дії. Редуктори відповідають за незмінне оновлення стану, гарантуючи, що початковий стан залишається незмінним.

Редукція сприяє односпрямованому потоку даних. Коли виконується дія, вона проходить через редуктори, які відповідно оновлюють стан. Будь-які компоненти, зацікавлені у зміні стану, можуть підписатися на сховище і отримувати оновлення щоразу, коли стан змінюється.

Для підключення компонентів до сховища Redux у React-додатках зазвичай використовується бібліотека під назвою "react-redux". Вона надає функцію підключення, яка дозволяє компонентам отримувати доступ до стану зі сховища та відправляти дії.

До переваг використання Redux включають такі фактори.

6. Централізоване управління станами. Redux централізує стан додатку, що робить його легшим для розуміння та управління. Це усуває потребу в розпорошеному управлінні станами в різних компонентах, що призводить до більш передбачуваного та зручного для підтримки коду.

7. Подорожі в часі та налагодження. Redux дозволяє здійснювати налагодження в часі, що означає, що розробники можуть відтворювати дії і бачити, як стан змінюється з часом. Ця функція є безцінною при діагностиці та виправленні помилок.

8. Масштабованість і повторне використання. Redux забезпечує масштабовану архітектуру, яка може працювати зі складними додатками з великими деревами станів. Це дозволяє краще використовувати код повторно, оскільки компоненти можуть отримати доступ до стану і відправляти дії без передачі пропсів через кілька рівнів.

Незважаючи на свої переваги, Redux може внести додаткову складність у невеликі або нескладні додатки. Він рекомендується для додатків, що потребують розширених функцій управління станами.

У процесі розробки інтернет-магазину доцільно використовувати хмарні backend-рішення, які дозволяють скоротити час розробки та спростити підтримку системи. Тому в якості серверної частини та бази даних обрано платформу Firebase, яка надає набір готових сервісів для створення повноцінних вебзастосунків [24]. Використання даного сервісу дозволяє значно спростити розробку backend-логіки. Використання Firebase є обґрунтованим з огляду на такі фактори:

- відсутність потреби розгортання та адміністрування власного сервера;
- швидка інтеграція бази даних і системи автентифікації;
- можливість роботи в реальному часі;
- масштабованість і висока доступність сервісів.

Це особливо важливо для інтернет-магазину, де необхідно забезпечити безперервну роботу системи, обробку замовлень та збереження даних

користувачів. Платформа Firebase має низку функціональних можливостей, які роблять її ефективною для розробки вебзастосунків:

1. Хмарна база даних. Використання Cloud Firestore дозволяє зберігати дані у вигляді документів JSON. Це забезпечує гнучкість структури та швидке отримання інформації без складних запитів (рис. 2.8).

2. Система автентифікації. Firebase Authentication підтримує реєстрацію користувачів за допомогою email, Google та інших сервісів, що значно спрощує реалізацію функцій входу в систему.

3. Хостинг і деплой. Firebase Hosting дозволяє швидко розгорнути вебзастосунок із високою швидкістю завантаження та підтримкою HTTPS.

4. Масштабованість. Сервіс автоматично масштабується залежно від навантаження, що є важливим для інтернет-магазину при зростанні кількості користувачів.

5. Інтеграція з іншими сервісами. Firebase легко інтегрується з frontend-технологіями, зокрема React, що забезпечує зручну взаємодію клієнтської та серверної частин [25].

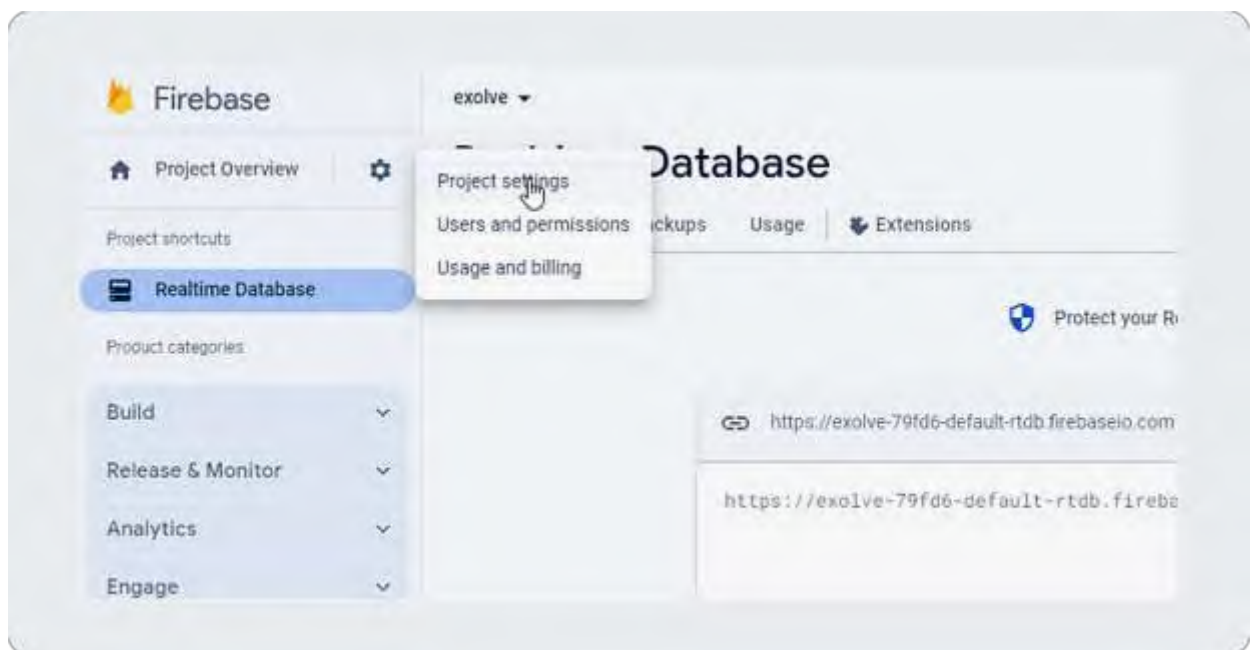


Рисунок 2.8 – Початок створення нового додатку в середовищі Firebase [24]

Інтеграція Firebase у вебзастосунок здійснюється у декілька етапів.

1. Створення проєкту: на платформі Firebase створюється новий проєкт, після чого отримуються конфігураційні параметри (API key, project ID тощо).
2. Підключення до клієнтського застосунку шляхом встановлення офіційної бібліотеки Firebase У проєкті на React: `npm install firebase`.
3. Ініціалізація Firebase: у коді створюється конфігураційний файл для підключення (рис. 2.9).

```
import { initializeApp } from "firebase/app";

const firebaseConfig = {
  apiKey: "...",
  authDomain: "...",
  projectId: "...",
};

const app = initializeApp(firebaseConfig);
```

Рисунок 2.9 – Запис скрипта для підключення Firebase

4. Далі підключаються необхідні модулі: Firestore – для роботи з базою даних; Authentication – для авторизації користувачів.
5. Використання у застосунку відбувається так: дані з бази отримуються через API Firebase, що дозволяє відображати каталог товарів, зберігати замовлення та обробляти дії користувача.

Для роботи з зображеннями обрано сервіс Cloudinary, який забезпечує зберігання, оптимізацію та швидке завантаження медіафайлів (рис. 2.10.)

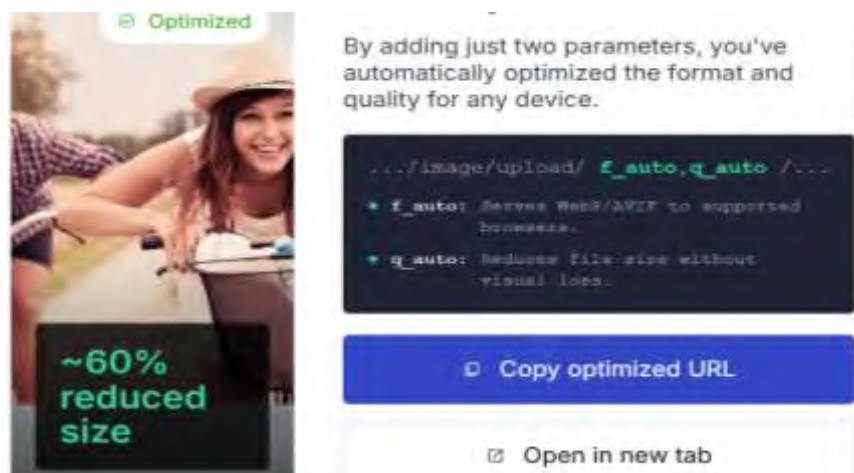


Рисунок 2.10 – Вигляд вікна для обробки зображень (тест) в Cloudinary [26]

На платформу Cloudinary можна розмістити власні зображення та обробити їх кількома способами, оптимізуючи розмір або інші властивості без втрат якості (див. рис. 2.9), що дозволяє розробникам надавати високоякісні медіа з оптимальним розміром файлів і часом завантаження. Вона покликана спростити процес зберігання, оптимізації, маніпулювання та доставки зображень і відео для веб- та мобільних додатків [27]. Cloudinary підтримує різноманітні формати зображень і відео, а також може обробляти зміну розміру, обрізання, обертання та інші перетворення на льоту, що полегшує адаптацію медіа-активів до різних пристроїв і розмірів екранів.

Cloudinary також надає потужний набір функцій для керування медіа-активами. Він пропонує безпечне зберігання, резервне копіювання та керування версіями, гарантуючи, що медіа-активи захищені, до них можна легко отримати доступ і відновити за потреби. Крім того, Cloudinary пропонує безперешкодну інтеграцію з популярними системами управління контентом (CMS) та фреймворками для розробки, що дозволяє легко інтегрувати можливості управління медіа в існуючі робочі процеси. Він надає набори для розробки програмного забезпечення (SDK) та API для різних мов програмування, що дозволяє розробникам інтегрувати функції Cloudinary у свої додатки та робочі процеси.

Використання Cloudinary підвищує продуктивність і зручність доставки медіа через свою глобальну мережу доставки контенту (CDN), незалежно від їхнього географічного розташування. Cloudinary має зручний вебінтерфейс і потужну консоль управління, що дозволяє розробникам і творцям контенту легко завантажувати, організовувати та керувати своїми медіа-активами. Він також пропонує функції аналітики та звітності, що дають уявлення про використання та продуктивність медіа.

Для початку роботи із сервісом достатньо зареєструватися. Cloudinary спрощує процес управління та доставки медіа-активів, пропонуючи комплексне рішення для розробників і творців контенту, а також надає необхідну документацію (наприклад, додаток А).

## 2.3 Розробка макетів окремих сторінок інтернет-магазину

Для початку необхідно створити орієнтовні макети різних сторінок. Макет інтернет-магазину повинен бути розроблений таким чином, щоб забезпечити безперебійний і візуально привабливий досвід перегляду. Він повинен бути добре організованим, з чіткою ієрархією, яка дозволяє користувачам легко знаходити інформацію та продукти, які вони шукають. Візуальні елементи, такі як зображення, кольори та типографіка, повинні використовуватися стратегічно, щоб покращити загальну естетику магазину та відобразити ідентичність бренду.

Макет головної сторінки буде містити слайдер – це елементи, зазвичай фото, які при певному інтервалі часу змінюються на наступний, блок для кого обирає речі клієнт, а саме чоловіки, жінки або діти та блок з посиланням на сторінку допомоги, де користувач зможе відправити листа (рис. 2.10).



Рисунок 2.10 – Вигляд макету головної сторінки

Макет сторінки з товарами буде містити різні фільтри такі як категорія, бренд, стать, колір, розмір, сортування тощо.

Макет сторінки входу або реєстрації міститиме поля для вводу e-mail, пароль (якщо реєстрація, то ще ПІБ), кнопку, кнопки «Вхід» або «Реєстрація» через Google або Facebook (рис. 2.11).



Рисунок 2.11 – Вигляд макету сторінки з формою входу або реєстрації

Товари можуть мати знижки, різну кількість кольорів, тому при наведенні буде відображатися основне й інше фото цього товару (рис. 2.12).



Рисунок 2.12 – Вигляд макету сторінки з товарами

Одним із основних атрибутів інтернет-магазину є кошик для обліку покупок. Макет сторінки кошика включає товари, які можна додати в улюблені або видалити, змінити кількість товару. Справа у вікні буде знаходитися форма,

яка рахує остаточну ціну і має кнопку для переходу до оформлення адреси та реквізитів (рис. 2.13).



Рисунок 2.13 – Вигляд макету сторінки кошика

Макет сторінки оформлення замовлення буде містити поле заповнення адреси для доставки та заповнення реквізитів (рис. 2.14).

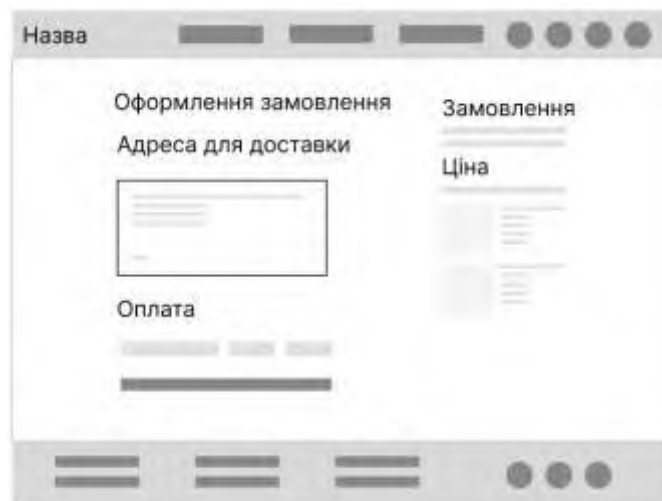


Рисунок 2.14 – Вигляд макету сторінки оформлення замовлення

Після успішного заповнення даних та натискання на кнопку, користувача повертатиме на головну сторінку. Завершивши розробку інтерфейсу та структури інтернет-магазину, можна приступити до написання коду.



## РОЗДІЛ 3

### ПРАКТИЧНІ АСПЕКТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІНТЕРНЕТ- МАГАЗИНУ СПОРТИВНОГО ОДЯГУ

#### 3.1 Технічні прийоми створення інтерфейсу головної сторінки

Головними напрямками дизайну інтерфейсу можна назвати засоби, які стали популярними завдяки технологіям і функціям CSS3, JS та програмам графічного дизайну.

Важливою частиною створення будь-якого сайту є семантична оптимізація коду. «Елементи програмного коду, які мають певне смислове значення, яке може сприйматися і запам'ятовуватися програмістом як осмислене ім'я, значення, зустрічаються в більшості мов програмування, а особливо в кодах розмітки вебсторінок HTML» [28].

Семантична оптимізація коду фокусується на використанні відповідних тегів HTML для відображення структури та ієрархії контенту. Це допомагає пошуковим системам зрозуміти контекст і релевантність інформації, що потенційно покращує видимість сайту в результатах пошукової видачі.

Використовуючи описові та змістовні імена для елементів HTML, розробники можуть зробити свій код більш зрозумілим і легким для сприйняття. Це не тільки приносить користь іншим розробникам, які можуть працювати над проектом, але й сприяє загальній легкості обслуговування кодової бази.

Ще одним ключовим аспектом семантичної оптимізації коду є розмежування проблем, пов'язаних із контентом, представленням і поведінкою. Такий підхід дозволяє розробникам підтримувати чітке розділення HTML, CSS і JavaScript, що полегшує оновлення та модифікацію певних аспектів вебсайту, не впливаючи на інші.

Для розробки головної сторінки створені 2 файли: перший це файл з розширенням «.js», який слугує для написання коду; інший файл із розширенням «.scss», який призначений для стилю різних елементів коду.

SCSS (Sassy CSS) – це препроцесор, який розширює можливості звичайного CSS. Він надає додаткові можливості, такі як змінні, вкладеність та інші, які роблять написання стилів більш ефективним та організованим [29].

Якщо брати до уваги інші файли, а саме зображення, то вони знаходяться в кореневій папці, у папці «images», де знаходяться всі зображення, крім зображень товарів, доступ до них йде через посилання в коді (рис. 3.1).

```
const img1 = require(`../../images/adidas-banner.jpg`);
const img2 = require(`../../images/nike-banner.jpg`);
const img3 = require(`../../images/puma-banner.jpg`);
```

Рисунок 3.1 – Вигляд посилання на зображення

Перше, що видно на головній сторінці, це верхня частина вебсайту або «шапка» (створюється як header). Після нього зображений слайдер із різними фото, який через певний проміжок часу змінює один елемент на інший (рис. 3.2).



Рисунок 3.2 – Вигляд слайдера на головній сторінці

При розробці слайдера був використаний фреймворк Swiper, який дозволяє використати готовий код для слайдерів.

Swiper – це найсучасніший безкоштовний мобільний сенсорний слайдер з апаратним прискоренням переходів і дивовижною нативною поведінкою.

Він призначений для використання на мобільних вебсайтах, мобільних вебдодатках і нативних/гібридних мобільних додатках [30]. Лістинг коду до слайдера показано в додатку Б.

Однією з ключових переваг Swiper є широкий набір опцій та функцій, що налаштовуються. Розробники мають тонкий контроль над поведінкою і зовнішнім виглядом слайдера за допомогою широкого спектру параметрів конфігурації. Вони можуть визначити кількість видимих слайдів, встановити ефекти переходу, налаштувати елементи керування навігацією та увімкнути автовідтворення, серед багатьох інших опцій. Така гнучкість дозволяє створювати індивідуальні та візуально привабливі слайдери, які відповідають конкретним вимогам до дизайну та функціональності проекту [31].

Swiper підтримує різні типи слайдів, включаючи зображення, HTML-контент, відео і навіть вкладені слайдери. Така універсальність дозволяє розробникам створювати складні та інтерактивні презентації з багатим мультимедійним та динамічним контентом.

Ще однією важливою особливістю Swiper є його можливості оптимізації продуктивності. Він використовує апаратне прискорення і розумні методи попереднього завантаження, щоб забезпечити плавні та ефективні переходи між слайдами, навіть великі обсяги контенту або зображення високої роздільної здатності. Після цього блоку на головній сторінці розміщений блок «Для кого обираєте?» (рис. 3.3).



Рисунок 3.3 – Вигляд блоку із вибором категорії користувачів

Тут розміщено три зображення з чоловіком, жінкою та дитиною, які посилаються на сторінку з товарами з відповідним фільтром по категорії покупців. Лістинг коду до рисунку 3.3 наведено на рис. 3.4.

```
<div className={style.for__whom__block}>
  <h2>ДЛЯ КОГО ОБИРАЄТЕ?</h2>
  <div className={style.for__whom__wrapper}>
    {forImages.map((item,i)=>{
      return(
        <Link key={i} to={headerTitle[i]}>
          <h2>{headerTitleUA[i]}</h2>
          <img src={item}></img>
        </Link>
      )
    })}
  </div>
```

Рисунок 3.4 – Вигляд коду для створення блоку із вибором категорії користувачів

Останній блок на головній сторінці це фото та форма з текстом та кнопкою, при натисканні на яку користувач переходить на сторінку підтримки (рис. 3.5).



Рисунок 3.5 – Вигляд блоку із посиланням на сторінку підтримки

Лістинг коду до рисунку 3.5 показано на рис. 3.6.

Для забезпечення читабельності сайту на різних моніторах, був розроблений адаптивний дизайн всьому сайту (рис. 3.7).

```

<div className={` ${style.support_block} ${style.desktop}`}>
  <div className={style.support_text}>
    <h1>Є питання або потрібна допомога?</h1>
    <Link to={"support"}>
      <Button text={"НАПИСАТИ"} />
    </Link>
  </div>
  <img src={support_big_img}/>
</div>
<div className={` ${style.support_block} ${style.mobile}`}>
  <img src={support_small_img}/>
  <div className={style.support_text}>
    <h1>Є питання або потрібна допомога?</h1>
    <Link to={"support"}>
      <Button text={"НАПИСАТИ"} />
    </Link>
  </div>
</div>

```

Рисунок 3.6 – Вигляд коду для створення блоку з підтримкою

Адаптивний дизайн спрямований на те, щоб вебсайти були візуально привабливими, незалежно від того, чи користувачі отримують доступ до них зі стаціонарного комп'ютера, ноутбука, планшета або смартфона [32].

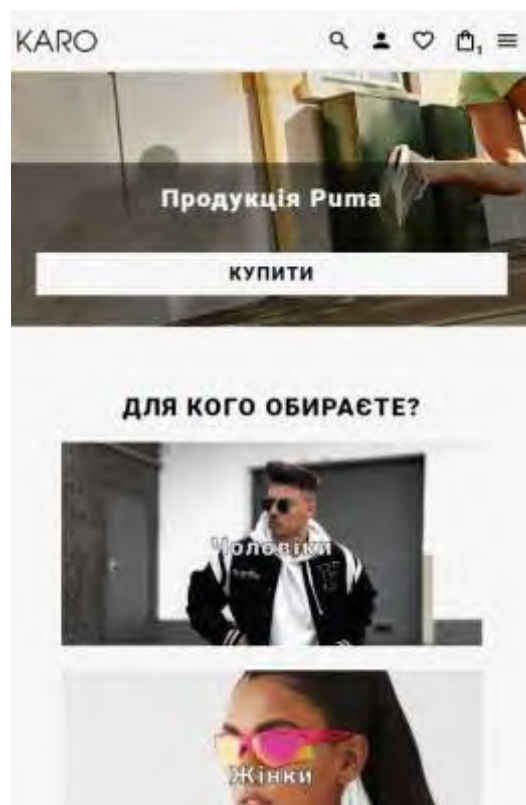


Рисунок 3.7 – Вигляд адаптивного дизайну головної сторінки

Отже, головна сторінка інтернет-магазину спортивного одягу має візуально привабливий і добре продуманий вигляд, який ефективно задовольняє потреби клієнтів, які були попередньо вивчені.

### 3.2 Обробка даних на сайті

Для інтернет-магазину важливо спроектувати базу даних, яка міститиме інформацію про товари, замовлення клієнта. Від цього залежить точність та продуктивність роботи застосунку. Firestore не використовує таблиці та зв'язки, як у реляційних СУБД. Дані організовані за ієрархією: Колекція → Документ → Підколекція → Документ. Це є документно-орієнтована NoSQL база даних, що зберігає інформацію у вигляді колекцій і документів. В такому виконанні колекція це аналог таблиці, документ – аналог запису. Кожен документ зберігає дані у форматі JSON. Для зручної і ефективної обробки даних інтернет-магазину створена панель для адміністрування, де можна додавати, редагувати та видаляти товари, переглядати замовлення та повідомлення (рис. 3.8).

Рисунок 3.8 – Вигляд сторінки зі створенням каталогу товарів

Після заповнення товару натискаємо кнопку «Create». Після цього дані про товар відправляються в базу даних, створену за технологією Firebase (рис.3.9).

```
products
└─ productId
  name: "Футболка PUMA Training"
  brand: "PUMA"
  price: 1299
  categoryId: "tshirts"
  sizes: ["S", "M", "L"]
  colors: ["black", "white"]
  imageUrl: "https://..."
  description: "Легка спортивна футболка..."
  instock: true
```

Рисунок 3.9 – Інформація про товар в Firebase

Посилання для зображень взято з платформи Cloudinary. Після оформлення замовлення на сайті, в базі даних Firebase віднімається кількість наявного товару на кількість замовленого товару (рис. 3.10). Якщо товару немає, користувач не зможе його вибрати. Колекція orders містить інформацію про замовлення (рис. 3.10).

```
orders
└─ orderId
  userId: "abc123"
  totalPrice: 2599
  status: "pending"
  createdAt: Timestamp
  items: [
    {
      productId: "p1",
      name: "Кросівки Adidas",
      quantity: 1,
      price: 2599
    }
  ]
```

Рисунок 3.10 – Інформація про оформлені замовлення в Firebase

Також після оформлення замовлення, це замовлення зберігається в окрему колекцію, де адміністратори сайту зможуть відстежити замовлення (рис. 3.11).



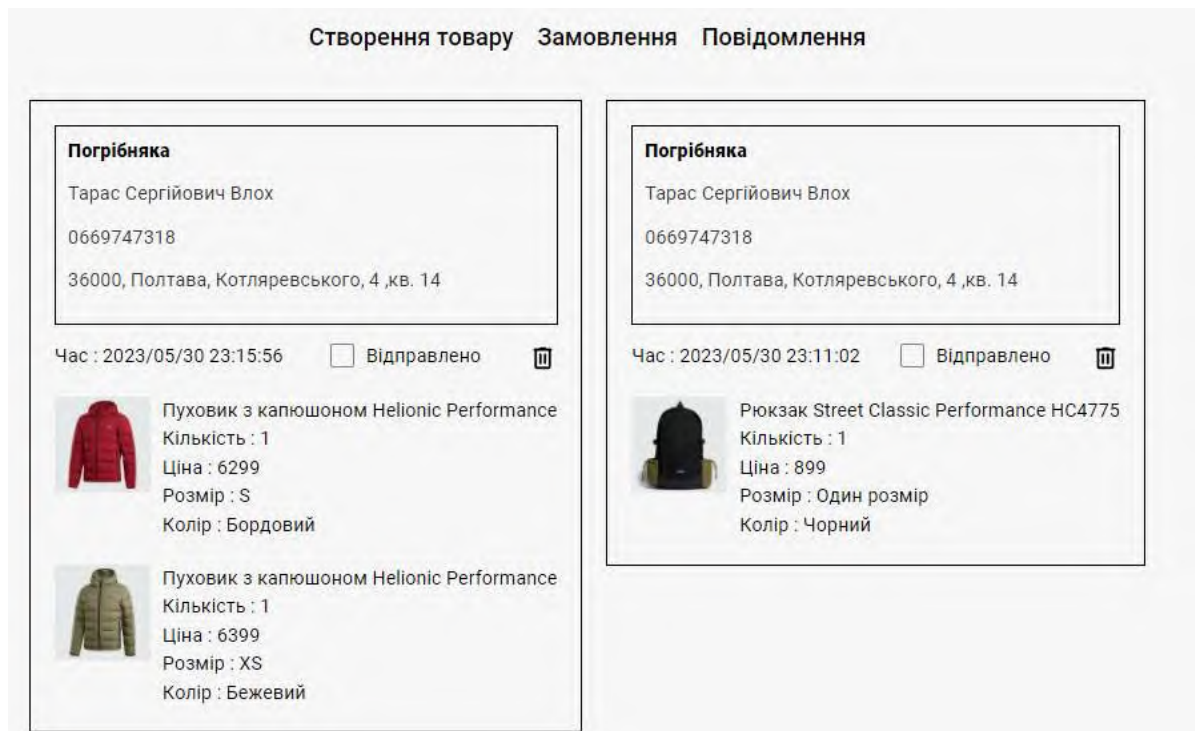


Рисунок 3.11 – Вигляд сторінки з відстеженням замовлення

Після того, як користувач відправив листа в підтримку, це повідомлення надходить до адміністраторів (рис. 3.12).

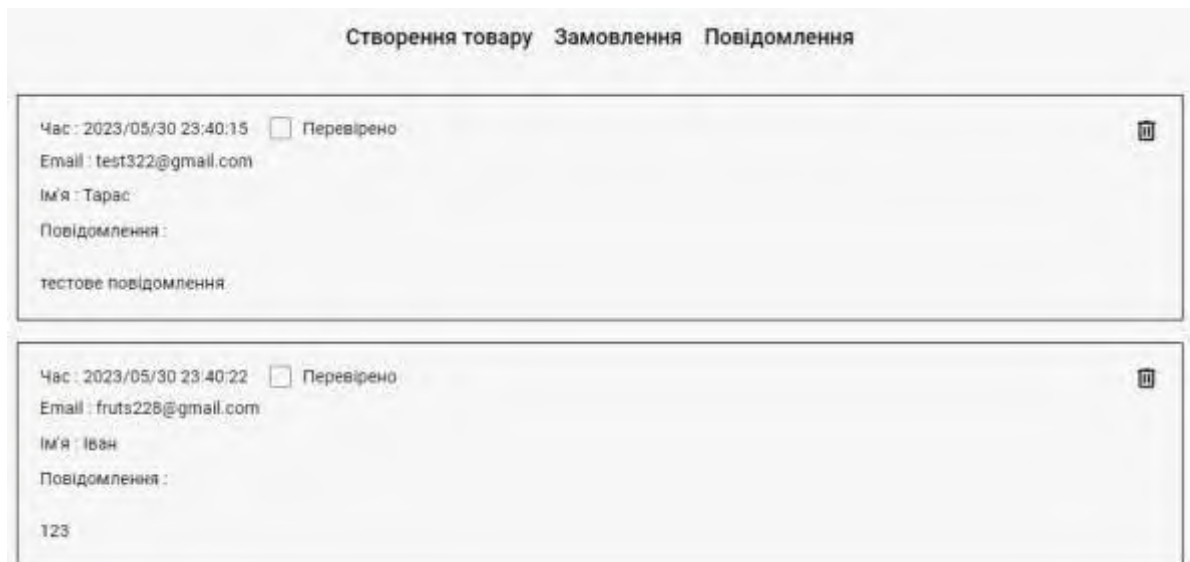


Рисунок 3.12 – Вигляд сторінки з повідомленнями

Сайт розроблений таким чином, що користувачу необов'язково реєструватися, щоб робити замовлення. Для користувачів, які зареєстровані, їхні



дані знаходяться в базі даних, це дає змогу паралельно отримувати дані з різних пристроїв. Для тих користувачів, які не зареєстровані, є одна особливість мови програмування JavaScript, а саме Local Storage, який зберігає інформацію тільки на цьому пристрої, певного браузера, свого роду локальна пам'ять.

LocalStorage – це функція веббраузера на JavaScript, яка дозволяє зберігати пари ключ-значення локально на пристрої користувача. Це простий і зручний спосіб постійного зберігання даних без необхідності взаємодії з сервером або внутрішньою базою даних [33].

LocalStorage є частиною API вебсховища, разом з sessionStorage, який пропонує схожу функціональність, але з різними масштабуванням і правилами життя. На відміну від sessionStorage, який зберігає дані протягом усього сеансу роботи в браузері, дані, що зберігаються в LocalStorage, зберігаються навіть після закриття і повторного відкриття браузера [34].

Щоб використовувати LocalStorage, можна отримати доступ до об'єкта localStorage, який надається об'єктом вікна браузера. Об'єкт localStorage діє як сховище ключів-значень, де ключі та значення зберігаються у вигляді рядків. Це означає, що будь-які нестрокові значення повинні бути перетворені в рядки перед збереженням, а при отриманні - розібрані до початкових типів. Нижче наведено приклад використання LocalStorage (рис. 3.13).

```
// Зберігання даних у LocalStorage
localStorage.setItem('key', 'value');

// Отримання даних з LocalStorage
const data = localStorage.getItem('key');
console.log(data); // Output: "value"

// Видалення даних з LocalStorage
localStorage.removeItem('key');
```

Рисунок 3.13 – Приклад використання LocalStorage

У наведеному вище прикладі, див. рис. 3.13, використано setItem() для зберігання значення, пов'язаного з ключем 'key'. Потім можна отримати значення

за допомогою `getItem()` і видалити його за допомогою `removeItem()`. `LocalStorage` також надає декілька додаткових методів. Властивість `localStorage.length` повертає кількість елементів, що зберігаються у `LocalStorage`, а метод `localStorage.clear()` видаляє всі пари ключ-значення зі сховища.

Зазвичай `LocalStorage` обмежена для зберігання близько 5 МБ даних, хоча точна межа може відрізнятись в різних браузерах. Важливо пам'ятати, що дані, які зберігаються в `LocalStorage`, доступні через код JavaScript, запущений на тому ж домені, а це означає, що він не підходить для зберігання конфіденційної інформації.

`LocalStorage` може бути корисним для різних цілей, наприклад, для зберігання користувацьких налаштувань, кешування даних або збереження даних, введених у формі, щоб забезпечити кращий користувацький досвід. Однак важливо використовувати `LocalStorage` розумно і пам'ятати про його обмеження, такі як обмеження на розмір сховища і той факт, що дані зберігаються локально на пристрої користувача.

Загалом, обробка даних на сайті вирішує всі проблеми, з якими може зіткнутися як користувач, так і адміністратор. Завдяки зручному інтерфейсу адміністратори можуть легко створювати, редагувати та видаляти товари, а також отримувати доступ до інформації про замовлення та сповіщень. Ця панель адміністрування підвищує продуктивність і спрощує процес контролю та підтримки роботи інтернет-магазину. Надаючи зручну та ефективну платформу, панель дозволяє адміністраторам ефективно вирішувати завдання обробки даних, забезпечуючи безперебійний та організований робочий процес.

### **3.3 Перевірка та опис економічної ефективності розробки**

Перевірка та опис економічної ефективності розробки проєкту передбачає детальну оцінку різних аспектів для визначення його життєздатності та ефективності у створенні економічних переваг.

Аналіз витрат. Для розглянутого проєкту розробки, що включає створення вебдодатку з використанням React, Firebase та Redux, були враховані наступні фактори витрат:

1. Інструменти та програмне забезпечення для розробки. Використані Visual Studio Code як редактор коду, який є безкоштовним і широко використовуваним інструментом у спільноті розробників. Він допоміг ефективно писати та керувати кодом.

2. Хостинг та інфраструктура. Firebase пропонує безкоштовний план хостингу, що дозволило розгорнути свій вебдодаток без жодних витрат на хостинг. Ця економічна опція забезпечила зручну та надійну платформу для розміщення додатку.

3. Сторонні бібліотеки та пакети. Ретельно відібрані бібліотеки та пакети з відкритим вихідним кодом для власного проєкту, що дозволило уникнути будь-яких витрат, пов'язаних з придбанням або ліцензуванням.

4. Витрати на персонал. Зазвичай, враховується час роботи задіяних у проєкті прикладних спеціалістів та оплата праці.

5. Витрати на інтеграцію та API. Розробляючи вебдодаток, необхідно врахувати будь-які потенційні витрати, пов'язані з інтеграцією сторонніх сервісів або використанням зовнішніх API. Залежно від конкретних вимог інтернет-магазину спортивного одягу, може виникнути потреба в інтеграції з платіжними шлюзами, постачальниками послуг доставки або іншими зовнішніми системами.

Інтеграція з цими сервісами часто пов'язана з додатковими витратами, такими як комісія за транзакції, підписка або плата за використання. Важливо оцінити моделі ціноутворення та умови цих інтеграцій, щоб переконатися, що вони відповідають бюджету та вимогам проєкту. З вище описаних витрат, могли бути витрати тільки за оплату за замовлення та комісію. Оскільки використовував імітацію оплати за замовлення, мені обійшлося це безкоштовно.

6. Обслуговування та підтримка. Хоча це не було безпосередньо пов'язано з етапом розробки, врахував потенційні майбутні витрати на

обслуговування та підтримку вебдодатку. Це включало такі фактори, як поточне обслуговування сервера, виправлення помилок і розширення функцій. Якщо використовувати проєкт для багатьох користувачів, то потрібно буде витрати кошти на базу даних.

Під час виконання кваліфікаційної роботи, використовуючи безкоштовні ресурси, такі як Visual Studio Code та безкоштовний хостинг Firebase, вдалося досягти ефективного керування витратами під час розробки власного вебдодатку. Ця економічна ефективність дозволила зосередитися у процесі розробки на успішному виконанні проєкту без фінансових обмежень.

Наприкінці додається приблизний економічний розрахунок на основі типових витрат на розробку інтернет-магазину у 2026 р. У розрахунках використовувався середній обмінний курс  $\approx 43$  UAH/USD на початок 2026 р. та поточні тарифи на хмарні послуги.

В таблиці 3.1 наведено орієнтовний розрахунок витрат, які складуть вартість розробки при замовленні в спеціалізованій компанії.

Таблиця 3.1 – Перелік витрат при розробці інтернет-магазину на замовлення бізнесу за розцінками станом на кінець 2025 р.

| № п/п | Найменування послуги (зміст)                                                                        | Оціночна вартість, грн |
|-------|-----------------------------------------------------------------------------------------------------|------------------------|
| 1     | Планування та дизайн. Збір вимог, структурування та UX дизайну                                      | 18 000,00              |
| 2     | Frontend розробка. HTML/CSS кодування, реалізація UI                                                | 45 000,00              |
| 3     | Backend розробка. Налаштування бази даних, логіка на стороні сервера                                | 40 000,00              |
| 4     | Інтеграція API (платіжний шлюз, тощо)                                                               | 15 000,00              |
| 5     | Тестування та QA. Забезпечення якості, виправлення помилок                                          | 12 000,00              |
| 6     | Впровадження та запуск. Налаштування сервера, реєстрація домену використовуючи Firebase             | 6 000,00               |
| 7     | Поточне обслуговування. Оновлення, виправлення помилок, технічна підтримка (терміном дії на місяць) | 8 000,00               |
| 8     | Сумарні витрати на розробку, встановлення та обслуговування інтернет-магазину                       | 144000,00              |

Обґрунтуємо показники, що наведені в табл. 3.1, за видами робіт. Планування і дизайн (позначимо  $B_{\text{диз}}$ ) включає: аналіз вимог, UX-схеми,

структуру каталогу товарів інтернет-магазину. Разом ці операції складають 40-50 годин роботи UX-дизайнера за тарифами 450 грн /год, тобто 18000 грн [36].

Найбільш трудомістка частина в частині frontend-розробки ( $B_{front}$ ) : каталог товарів, фільтри, кошик, адаптивність. Разом обсяг робіт орієнтовно 100–120 годин, відповідно сумарна вартість роботи від 45000 грн.

Виконання завдань Backend-частини ( $B_{back}$ ) включає: моделювання структури даних, налаштування бази, обробку замовлень. При використанні Firebase це дешевше, ніж класичний сервер. Firebase має безкоштовні квоти, після яких оплата залежить від трафіку та запитів. Для студентського або малого магазину: перші місяці – 0–500 грн/міс.

Cloudinary має безкоштовний тариф, або \$99/міс для розширених можливостей. В роботі для розрахунку прийнято базовий тариф з безкоштовним використанням.

Повну вартість розробки можна записати за формулою, яка включає всі види витрат:

$$B = B_{div} + B_{front} + B_{back} + B_{api} + B_{qa} + B_{deploy} + B_{sup} \quad (3.1)$$

Підставляючи у формулу 3.1 наведені розрахунки та дані з табл. 3.1, отримуємо сумарну вартість 144 тис. грн.

Проведений економічний розрахунок показав, що орієнтовна вартість розробки інтернет-магазину спортивного одягу з використанням сучасних вебтехнологій становить близько 144 тис. грн за розцінками робіт і тарифами станом на кінець 2025 р. Найбільшу частку витрат становить frontend та backend-розробка, що пояснюється високою трудомісткістю створення користувацького інтерфейсу та серверної логіки. Використання хмарних сервісів, зокрема Firebase та Cloudinary, дозволяє зменшити витрати на інфраструктуру та прискорити процес впровадження системи.

Навіть, якщо етап розробки пройшов гладко, слід визнати, що в майбутньому можуть виникнути непередбачувані технічні проблеми. Ці

проблеми можуть включати проблеми сумісності між різними версіями бібліотек або потенційні вразливості безпеки, які можуть з'явитися. Щоб зменшити ці ризики, необхідно залишатися в курсі останніх версій і обирати сумісні інструменти. Тобто, продовжувати процес навчання та моніторингу вебтехнологій, а також збагачувати практичний досвід.

В результаті цих зусиль досягнуто хороших результатів у власному проєкті розробки. Додаток було розроблено ефективно, з використанням економічно вигідних ресурсів та з дотриманням найкращих практик. Прорахувавши потенційні ризики, був створений безпечний, придатний до масштабування й зручний вебдодаток.

Загалом, поєднання використання економічно ефективних ресурсів, проведення ретельної оцінки ризиків та впровадження відповідних заходів сприяло успішному завершенню проєкту з розробки якісного інтерфейсу користувача інтернет-магазину.

## ВИСНОВКИ

У процесі виконання кваліфікаційної роботи на основі здійсненого аналізу було проаналізовано особливості розвитку електронної комерції та сучасних інтернет-магазинів, зокрема у сфері продажу спортивного одягу, принципу розробки інтернет-магазину спортивного одягу можна зробити наступні висновки на завершення роботи.

1. У теоретичній частині було досліджено принципи функціонування інтернет-магазинів, а також проведено порівняльний аналіз існуючих рішень на прикладі вебсайтів провідних брендів спортивного одягу. Аналіз інтерфейсів та функціональних можливостей дозволив визначити ключові характеристики ефективного вебресурсу: зручна навігація, якісне представлення товарів, адаптивний дизайн та оптимізований процес обробки замовлень.

2. На основі аналізу сучасних платформ електронної комерції було сформовано функціональні та нефункціональні вимоги до розроблюваного інтернет-магазину. Було визначено, що система повинна забезпечувати управління каталогом товарів, обробку замовлень, інтеграцію платіжних та логістичних послуг, а також відповідати вимогам щодо продуктивності, безпеки та доступності.

3. У практичній частині було обрано сучасний стек вебтехнологій, зокрема, бібліотеку React для реалізації користувацького інтерфейсу, інструмент керування станом Redux, хмарну платформу Firebase для зберігання та автентифікації даних, а також сервіс Cloudinary для роботи з медіаконтентом. Середовищем розробки був Visual Studio Code, який надає зручні інструменти для створення та тестування вебдодатків.

4. Структура бази даних була розроблена на основі документоорієнтованої моделі Cloud Firestore, яка включає основні сутності інтернет-магазину: користувачів, товари, категорії та замовлення. Такий підхід дозволяє забезпечити масштабованість системи, швидке отримання даних та спрощення логіки сервера.

5. Практичні аспекти розроблення інтернет-магазину спортивного одягу показали, що створення візуально привабливого та інтуїтивно зрозумілого користувацького інтерфейсу є життєво важливим для покращення досвіду перегляду та залучення клієнтів.

6. Орієнтовна вартість розробки інтернет-магазину спортивного одягу з використанням сучасних вебтехнологій становить близько 144 тис. грн станом на початок 2026 р. Найбільшу частку витрат становить frontend та backend-розробка, що пояснюється високою трудомісткістю створення користувацького інтерфейсу та серверної логіки.

Враховуючи та впроваджуючи теоретичні основи розробки інтернет-магазинів, обираючи відповідні методи та звертаючись до практичних аспектів, розробка інтернет-магазину спортивного одягу виконана успішно. Ці висновки забезпечують міцну основу для створення надійної та зручної онлайн-платформи для продажу спортивного одягу. Отримані результати можуть бути використані як основа для подальшого розширення функціональності системи та її впровадження в практичну діяльність.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Laudon K. C., Traver C. G. *E-commerce 2021: Business, Technology, Society*. 16th ed. New York : Pearson Education, 2021. 912 p.
2. NetMarket. URL: <https://en.wikipedia.org/wiki/NetMarket> (дата звернення: 14.05.2026).
3. The History of E-commerce. URL: <https://www.retailspacesolutions.com/blog/the-history-of-ecommerce/> (дата звернення: 14.05.2026).
4. Червона О. Ю. Тенденції розвитку електронної комерції. *Науковий вісник Херсонського державного університету*. Серія «Економічні науки». 2020. Т. 39. С.65-68.
5. Chaffey D. *Digital Business and E-Commerce Management*. 7th ed. Harlow : Pearson Education, 2019. 704 p.
6. Інтернет магазин на wordpress: що потрібно знати перед створенням? *Хостинг Україна*: вебсайт. URL: <https://www.ukraine.com.ua/uk/blog/site-administration/internet-magazin-na-wordpress-chto-nuzhno-znat-pered-sozdaniem.html> (дата звернення: 14.05.2026).
7. Що таке безголова комерція? *Brainberry*: вебсайт. URL: <https://brainberry.ua/uk/newsroom/blog/entrepreneurs-guide-to-headless-commerce> (дата звернення: 14.05.2026).
8. Копішинська О. П., Уткін Ю. В., Скриль В.К. Особливості технологій і архітектури вебсервісів. *Інтеграція інформаційних систем і інтелектуальних технологій в умовах трансформації інформаційного суспільства*: Матер. IV Міжнар. науково-практичної конференції, Полтава, 21-22 жовтня 2021 року. Полтава: ПДАУ, 2021 С.29-33.
9. What Is Shopify and How Does It Work? What is Shopify? *Shopify*: вебсайт. URL: <https://www.shopify.com/blog/what-is-shopify> (дата звернення: 14.05.2026).
10. Мироненко М. Концептуально-методологічний підхід до розробки інтернет-магазину. *Сучасні інформаційні технології та інноваційні методики в*

*економіці, менеджменті та бізнесі*: Матеріали XXI щорічної студ. наукової конференції, м. Полтава, ПДАУ, 22 квітня 2026 р. Полтава: ПДАУ, 2026 р. С. 83-85.

11. The most flexible ecommerce platform. URL: <https://woocommerce.com/> (дата звернення: 14.05.2026).

12. Screenshot of the data collected by Shopify. *Pixartprinting*: вебсайт. URL: <https://www.pixartprinting.co.uk/blog/shopify-how-it-works/> (дата звернення: 14.05.2026).

13. *Adidas*: вебсайт. URL: <https://www.adidas.ua/> (дата звернення: 14.05.2026).

14. *Puma*: вебсайт. URL: <https://ua.puma.com/uk/> (дата звернення: 14.05.2026).

15. *Skechers*: вебсайт. URL: <https://www.skchrs.com.ua/> (дата звернення: 14.05.2026).

16. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення: 14.05.2026).

17. Using Git source control in VS Code. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення: 14.05.2026).

18. IntelliSense. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення: 14.05.2026).

19. Personalize Visual Studio Code. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення: 14.05.2026).

20. JavaScript-бібліотека для створення користувацьких інтерфейсів. *React*: вебсайт. URL : <https://uk.reactjs.org/> (дата звернення: 14.05.2026).

21. Вступ в React, якого нам не вистачало. Чому React? URL: <https://devzone.org.ua/post/vstup-v-react-yakogo-nam-ne-vistachalo> (дата звернення: 14.05.2026).

22. *Redux*: вебсайт. URL: <https://redux.js.org/> (дата звернення: 14.05.2026).

23. A reduced explanation of Redux. (Reducers, Actions, and Store). URL: <https://levelup.gitconnected.com/a-reduced-explanation-of-redux-reducers-actions-and-store-9b7819c5d064> (дата звернення: 14.05.2026).

24. Make your app the best it can be. *Firebase*: вебсайт. URL: <https://firebase.google.com/> (дата звернення: 14.05.2026).

25. Pricing plans. *Firebase*: вебсайт. URL: <https://firebase.google.com/pricing> (дата звернення: 14.05.2026).

26. *Cloudinary*: вебсайт. URL: <https://console.cloudinary.com/> (дата звернення: 04.03.2026).

27. Get started with Digital Asset Management. *Cloudinary*: вебсайт. URL: [https://cloudinary.com/documentation/digital\\_asset\\_management\\_overview](https://cloudinary.com/documentation/digital_asset_management_overview) (дата звернення: 30.03.2026).

28. Копішинська О. П., Гуйва О. О. Сучасні тенденції дизайну вебзастосовань. Development of modern science: theory, methodology, practice: Abstracts of XIII International Scientific and Practical Conference, Madrid, Spain March 18–19, 2021. P.209-211. URL: <http://dspace.pdaa.edu.ua:8080/handle/123456789/10429> (дата звернення: 14.05.2026).

29. Golub V., Kopishynska O. Analysis of semantic meanings of program code elements in the context of web application optimization. *Інтеграція інформаційних систем і інтелектуальних технологій в умовах трансформації інформаційного суспільства*: Матер. IV Міжнар. науково-практичної конференції, Полтава, 21-22 жовтня 2021 року. Полтава: ПДАУ, 2021. С.37-38.

30. *Sass*: вебсайт. URL: <https://sass-lang.com/> (дата звернення: 17.05.2026).

31. The Most Modern Mobile Touch Slider. *Swiper*: вебсайт. URL: <https://swiperjs.com/> (дата звернення: 30.03.2026).

32. Interface SwiperOptions. *Swiper*: вебсайт. URL: [https://swiperjs.com/types/interfaces/types\\_swiper\\_options.swiperoptions](https://swiperjs.com/types/interfaces/types_swiper_options.swiperoptions) (дата звернення: 30.03.2026).

33. Бондар, Д. С. Проєктування та розробка web-сайту «Посібник з HTML5». 2022. 20 с.

34. localStorage in JavaScript: A complete guide. URL: <https://blog.logrocket.com/localstorage-javascript-complete-guide/> (дата звернення: 30.03.2026).

35. Web Storage concepts and usage URL: [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Storage\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API) (дата звернення: 30.03.2026).

36. Скільки заробляє UI/UX дизайнер в Україні: огляд зарплат. *Mate.Academy*: вебсайт. URL: <https://mate.academy/blog/ui-ux-design/skilky-zarobliaie-uiux-dyzainer-v-ukraini-ohliad-zarplat> (дата звернення: 05.05.2026).